Hyperpriorsensitivity of Bayesian Wrapped Gaussian Processes with an Application to Wind Data

Master Thesis submitted

to

Prof. Dr. Nadja Klein

Humboldt-Universität zu Berlin

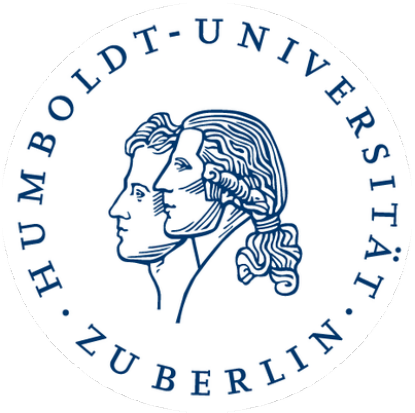
School of Business and Economics

Chair of Statistics

Prof. Dr. Thomas Kneib

Georg-August-Universität Göttingen

Chair of Statistics and Econometrics



by

Anna Elisabeth Riha

(583443)

in partial fulfillment of the requirements

for the degree of

Master of Science in Statistics

23 November 2020

Acknowledgements

Firstly, I want to thank my supervisor Nadja Klein for pointing me to the interesting world of Gaussian processes and spatial statistics in a Bayesian framework.

I would like to express gratitude to the open source communities around `R` (R Core Team 2020), `RStudio` (RStudio Team 2019) and `RMarkdown` (Allaire et al. 2020) and their exchange of knowledge and advice via GitHub, StackExchange and other means. Thanks to a wide range of available `R`-packages and especially the packages in the `tidyverse` (Wickham 2019) as well as `knitr` (Xie 2020b) and `bookdown` (Xie 2020a), writing and typesetting this thesis as well as analyzing data, running simulations and creating visualizations was very much enjoyable and insightful. Special thanks goes to Phi Nguyen and his project `huwiwidown` (Nguyen 2020) which provides a great `bookdown`-template following the guidelines of the Chair of Statistics at Humboldt-Universität zu Berlin.

I wish to thank Andreas Brandmaier for an inspiring and motivating time as a student assistant in his Formal Methods group at Max-Planck Institute of Human Development that continuously fostered my fascination for applied statistical modeling and machine learning methods as well as my former colleague Aaron Peikert for sharing his thoughts and ideas for reproducible workflows and a parallel simulation setup as in Peikert (2019). Thomas Siskos, Arndt Großmann and Elmar Csaplovics provided helpful comments on a first draft.

Lastly, I am grateful for the support of my friends, my family and a wonderful partner with an open mind, heart and arms.

Abstract

Directional or periodic measurements play a vital role in many scientific fields. Developing ways to analyze directional data while accounting for their specific characteristics is an important topic of ongoing research in the discipline of directional statistics. In addition to accounting for the directional nature of given measurements, applications like the meteorological analysis of wind directions face structured dependence of observations over space and time that needs to be accounted for.

This work presents methods to analyze directional data with such spatial and temporal dependence structures in a Bayesian framework using Gaussian processes. The focus is on wrapped spatial Gaussian process models as developed by Jona Lasinio, Gelfand, and Jona Lasinio (2012) and implemented in the `R`-package `CircSpaceTime` by G. Jona Lasinio, Mastrantonio, and Santoro (2019).

Hyperpriorsensitivity of such wrapped models is inspected with simulated data by comparing predictive performance of wrapped Gaussian process models for different hyperparameters in the hyperprior settings of the models implemented in `CircSpaceTime` while controlling for different training and test sample sizes. Spatial interpolation accuracy assessed by circular average prediction error (APE) and circular ranked probability scores (CRPS) emphasizes the central role of the spatial decay parameter ρ and the hyperparameter settings in its Uniform prior distribution.

An application of wrapped spatial Gaussian process models is presented for wind direction data from the German Weather Service (Deutscher Wetterdienst (DWD)) further exploring prediction accuracy for different test locations and different hyperprior settings. Results suggest that specific characteristics of a chosen date, like circular dispersion of the observed wind directions, have more influence on prediction accuracy than differing test locations or hyperparameter assumptions.

Contents

List of Abbreviations

APE	Average Prediction Error
CRPS	Continuous Ranked Probability Score
DWD	Deutscher Wetterdienst (German Weather Service)
GP	Gaussian Process
LEQR	Humboldt Lab of Empirical and Quantitative Research
MCMC	Markov Chain Monte Carlo
r.v.	random variable
UTM	Universal Transverse Mercator

List of Notations

`alpha`	circular mean in `CircSpaceTime`
c	concentration of a wrapped Gaussian obtained via $c = e^{-\sigma^2/2} < 1$
$\mathcal{D}$	training data
$\mathbf{f}$	vector of function values, training outputs
$\mathbf{f}_\star$	vector of unknown outputs
$g(\theta)$	probability density function of wrapped r.v. Θ with realization θ
$G_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$	Gram matrix defined by $\kappa(.,.)$
$\mathcal{GP}$	Gaussian process
γ_1, γ_2, γ_3	arbitrary angles measured in degrees
$h(y)$	probability density function of linear continuous r.v. Y with realization y
$H(y)$	probability distribution function of Y
h_{sp}	distances between spatial locations
K	winding number for the wrapping approach with realization k
$\mathbf{K}$	vector of winding numbers with realization $\mathbf{k}$
`k`	winding number in `CircSpaceTime`
$\kappa(.,.)$	kernel function or covariance function of a Gaussian process
$m(.)$, $\mu(.)$	mean function of a Gaussian process
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\omega(\mathbf{s})$	univariate spatial Gaussian process with mean zero
$\varphi(z)$	probability density function of $Z \sim \mathcal{N}(0,1)$
$\boldsymbol{\Psi}_Y$	a collection of parameters $(\mu_Y \mathbb{1}_n, \sigma_Y^2, \boldsymbol{\rho})^T$
$R(h_{sp}; \rho)$	(spatial) correlation function with spatial decay parameter ρ
ρ	spatial decay parameter
`rho`	spatial decay parameter in `CircSpaceTime`
$\{\mathbf{s}_i : i = 1, ..., n\}$	spatial locations $\mathbf{s}_i \in \mathcal{S} \subset \mathbb{R}^d$ with $d = 2$ in the spatial setting
$\mathbf{S}$	the covariance matrix of the training data $\mathbf{X}$
`sigma2`	variance parameter in `CircSpaceTime`
Θ	circular r.v. with realization θ
$\boldsymbol{\Theta} = (\Theta_1, ..., \Theta_p)^T$	a vector of circular r.v.'s
$\Theta(\mathbf{s})$	wrapped spatial GP
$\mathcal{WN}(\mu, c)$	wrapped Gaussian distribution with mean μ and concentration c
$\mathcal{X}$	domain of input data
$\mathbf{X}$	a matrix of given input data with $\mathbf{x}_i$ denoting the i-th input vector
$\mathbf{X}_\star$	a matrix of newly observed inputs
Y, Y_1, Y_2	linear (i.e. unwrapped) random variables
$Y(\mathbf{s})$	spatial linear (i.e. unwrapped) GP
$\mathbf{Y} = (Y_1, ..., Y_p)^T$	vector of linear r.v.'s, p realizations from a Gaussian process

List of Figures

List of Tables

1 Introduction

Directional data occurs when measuring directions and examining orientations of objects in a given coordinate system. Some examples are movements of animals, orientation of magnetic field lines or wind directions. Moreover, directional data can also describe periodic measurements like arrival or treatment times measured on a clock or periodically recurring events like daily, weekly or monthly biological cycles.
Thus, the analysis of directional data plays an important role in a wide range of research fields such as meteorology, astronomy, earth and environmental sciences, biology, psychology and behavioral and political sciences as illustrated in more detail in the first chapter of Mardia and Jupp (2000) or in Jammalamadaka and SenGupta (2001) as well as Ley and Verdebout (2017) and Ley and Verdebout (2019) for more recent modeling approaches and applications of directional statistics.

Following Jammalamadaka and SenGupta (2001) and Ley and Verdebout (2017), directional data denotes both data on a circular or spherical domain. Circular data refers to data measured on a circle, whereas spherical data describes data in $\mathbb{R}^d$ with dimensions $d \geq 3$.
As Ley and Verdebout (2017) point out concisely, the central challenge when working with directional data comes down to "the curvature of the sample space since the unit hypersphere or circle is a non-linear manifold" (Ley and Verdebout (2017), p. 1).
A simple example taken from Lee (2010) illustrates this. The average from a set of three angles with $\gamma_1 = 270°$, $\gamma_2 = 10°$ and $\gamma_3 = 20°$ would be wrongly calculated as $100°$ when summing the values and dividing by the number of occurences. As the linear mean formula does not take the circular nature of the data into account, i.e. that $0° \widehat{=} 360°$ or equivalently $0 \widehat{=} 2\pi$ on the circle, the average direction has to be computed using circular mean formulas like $\bar{\gamma}_{circ} = \arctan^* \left[\frac{\sum_{i=1}^{n} sin(\gamma_i)}{\sum_{i=1}^{n} cos(\gamma_i)} \right]$ as suggested by Ley and Verdebout (2017). For this set of angles the mean direction is actually roughly $19.22°$.

This can be illustrated further following an introductory example in Coles (1998). Figure 1 reproduces this example for a time series of 10-minute measurements of wind directions in radians observed at the weather station in Berlin-Tempelhof in March 2019 and available from the German Weather Service (DWD) (see Coles (1998) for a similar depiction of wind directions in the United Kingdom).
Figure 1 presents several occasions where the time series of wind directions changes abruptly showing large fluctuations. In fact, every time the wind direction measurements on the y-axis

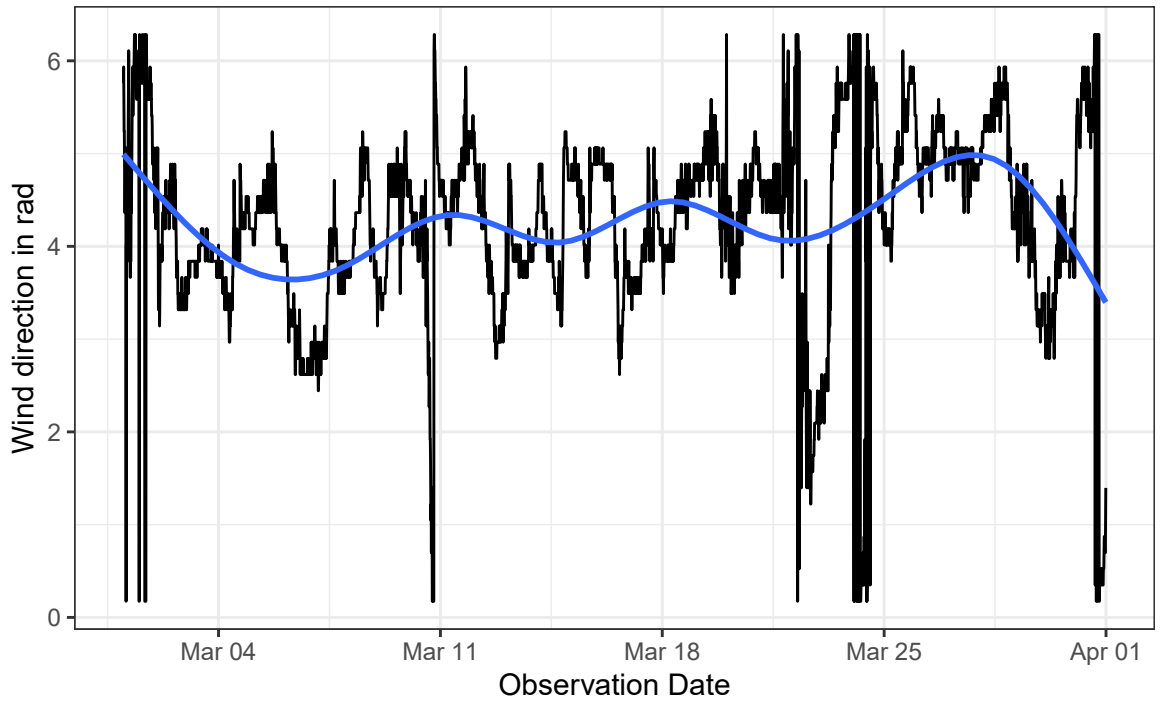


Figure 1: Wind directions at Berlin-Tempelhof weather station in March 2019 as 10-minute measurements in radians. Data taken from the German weather service (DWD).

reach 0 or $2\pi = 6.2831853$ respectively, the linear scale indicates large differences in wind directions. Only when changing the perspective to measurements on a circular domain, it becomes obvious that wind directions at these points in time are actually very similar. This illustrates how a linear scale fails to capture a circular variable correctly.

Jammalamadaka and SenGupta (2001) further motivate the importance of modeling tools that take directional characteristics into account by pointing out that e.g. directions in two dimensions can differ dependent on the selected rotation and origin and, due to the periodicity, have no ranking and require circular distance metrics.

In summary, the above examples emphasize that statistical modeling tools need to account for the directional characteristics of data measured on a circular or spherical domain.

Naturally, there is a broad occurrence of directional measurements when observing spatial or periodic phenomena in the physical sciences and elsewhere. Fisher (1993) insightfully explains how early measurements of earth's magnetism illustrate that "the roots of circular data analysis [reach] back [...] as far as the mid-18th Century" (Fisher (1993), p. 2).

Ley and Verdebout (2017) highlight the impact of Fisher (1953), who studied paleomagnetic data from lava flows in Iceland accounting for the directionality of the data, on a lot of subsequent efforts in finding directional counterparts for linear methods and approaches. In

his work, Fisher (1953) aimed at developing a theory of errors for spherical measurements posing the driving question of how a theory of measurement errors would be different "if the observations under discussion had in fact involved errors so large that the actual topology had had to be taken into account" (Fisher (1953), p. 295).

Other early examples of directional approaches are Watson (1961) who looked at goodness-of-fit tests for circular variables, Stephens (1963) who worked with random walks on the circle and Kent (1978) who presented asymptotic results for the von-Mises-Fisher distribution. In physics e.g. Breitenberger (1963) worked on circular distributions analogues of the Gaussian distribution. For an overview of these early developments see Fisher (1993). Circular distributions can be defined in different ways and Mardia and Jupp (2000) and Jammalamadaka and SenGupta (2001) present a wide range of examples of different circular distributions as well as results for inference with directional data.
The emergence of Markov chain Monte Carlo (MCMC) methods led to an increase in inferential analysis of directional data (see Ley and Verdebout (2017)). Examples are linear models for circular data as introduced in Fisher (1993) as well as Breckling (1989) who presented circular time-series analysis and Coles (1998) showing how MCMC methods allow fitting wrapped stochastic process models to circular data.

Mardia and Jupp (2000) and Jammalamadaka and SenGupta (2001) provide an extensive overview of the emergence of circular methods and the theoretical development of directional data analysis. Jona Lasinio, Gelfand, and Jona Lasinio (2012) and Mastrantonio, Gelfand, and Jona Lasinio (2016) also offer more insight in the historical development of inferential methods for circular data in their introductory remarks.

Current theoretical developments and research in the field of directional statistics is presented in Ley and Verdebout (2017) with a special focus on asymptotic statistics and high-dimensional problems. Moreover, Ley and Verdebout (2019) bring together recent applied directional data analysis e.g. directional models for protein structure in biology, spatial directional modeling, directional data analysis in the social and behavioral sciences and in machine learning setups. Like in other fields of statistics, the combination of machine learning frameworks, increasing computational power and availability of larger datasets is additionally pushing the developments in directional methods (see Ley and Verdebout (2017) and Ley and Verdebout (2019)).

Apart from these recent efforts, G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) argue in Ley and Verdebout (2019) that directional methods for observations from spatial and spatio-temporal processes are still thin on the ground.
Using a special class of stochastic processes, namely Gaussian processes, approaches in Jona Lasinio, Gelfand, and Jona Lasinio (2012), Mastrantonio, Jona Lasinio, and Gelfand (2016) and Mastrantonio, Gelfand, and Jona Lasinio (2016) developed tools for modeling and predicting circular spatial data in a Bayesian framework that allow for different parametrizations and prediction in a space of possible functions which is especially useful when looking at directions in space or time or both with an assumed dependence between locations and across time. Broadly speaking, these approaches work with spatial linear Gaussian processes and extend them to circular data by wrapping or projecting transformations.

This thesis focuses on wrapped spatial Gaussian processes for modeling and prediction of circular spatial data as developed in Jona Lasinio, Gelfand, and Jona Lasinio (2012), implemented in the `R`-package `CircSpaceTime` (G. Jona Lasinio, Mastrantonio, and Santoro 2019) and extended and further explained in Mastrantonio, Jona Lasinio, and Gelfand (2016), Mastrantonio, Gelfand, and Jona Lasinio (2016), G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) and Jona Lasinio, Santoro, and Mastrantonio (2020).

The next part introduces linear Gaussian processes and their extension to circular spatial setups in a Bayesian framework with the wrapping transformation from Jona Lasinio, Gelfand, and Jona Lasinio (2012).
A simulation study in section 3 tests the hyperprior sensitivity of these models for different hyperparameters in the prior distributions of the wrapped Gaussian process models implemented in `CircSpaceTime` (G. Jona Lasinio, Mastrantonio, and Santoro 2019).
Section 4 presents an application for spatial interpolation of wind directions in a dataset from the German Weather Service (Deutscher Wetterdienst (DWD)) comparing predictive accuracy for different dates, test locations and hyperprior settings. Section 5 concludes with a summary and questions of interest for future research.

2 Methods: Gaussian Processes and Directional Data

As introduced, analyzing directional data is a topic of interest in many research enterprises. Tools that account for the properties of directional data while at the same time allowing for explicit modeling of spatial and temporal dependence structures are of special interest in

fields like geostatistical data analysis and spatial statistics.

Therefore, stochastic processes and especially Gaussian processes (GPs) are a widely used modeling tool in spatial statistics as they allow modeling collections of random variables with dependence structures in space or time or both. As Diggle and Ribeiro jr. (2007) summarize, Gaussian processes are a common tool in geostatistics as they provide a "flexible, empirical model for an irregularly fluctuating, real-valued spatial surface" (Diggle and Ribeiro jr. (2007), p.13).

Starting from a linear spatial Gaussian process, this section presents how Gaussian processes can be extended to circular data with a wrapping transformation as developed in Jona Lasinio, Gelfand, and Jona Lasinio (2012) and further explained and extended by Mastrantonio, Jona Lasinio, and Gelfand (2016), Mastrantonio, Gelfand, and Jona Lasinio (2016), G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) and Jona Lasinio, Santoro, and Mastrantonio (2020).

2.1 Introducing Gaussian Processes

In general, stochastic processes describe indexed collections of random variables with a dependence structure over an index set e.g. points in time or space. As introduced by Rasmussen and Williams (2006), Gaussian processes are a type of stochastic processes with the defining property that any finite subset of realizations of this process follows a joint multivariate Gaussian distribution. Following the definition of a multivariate Gaussian distribution, this means that any finite subset, $\mathbf{Y} = (Y_1, ..., Y_k)^T$, of k realizations of the Gaussian process follows a k-variate multivariate Gaussian distribution. This in turn implies that any linear combination $a_1Y_1 + a_2Y_2 + ... + a_kY_k$ for any constant vector $a \in \mathbb{R}^k$ follows a univariate Gaussian distribution (see Härdle and Simar (2019) for an overview of the multivariate Gaussian and its properties).

There exists a wide range of literature on Gaussian processes. For example, Rasmussen and Williams (2006) offer an in-depth introduction to Gaussian processes as well as their fundamental building blocks like covariance functions and illustrate their usefulness in regression and classification questions in machine learning applications. Also, chapter 15 in Murphy (2012) provides further introduction and discussion of Gaussian process methods in the context of machine learning. Görtler, Kehlbeck, and Deussen (2019) offer an intuitive explanation of the core elements of Gaussian processes and provide insightful visualizations on the website of their article. Gaussian processes and their wide range of applications in

spatial statistics are presented in great detail in Diggle and Ribeiro jr. (2007).
The focus of this section will ultimately be on the extensions of Gaussian processes to circular spatial data as developed in Jona Lasinio, Gelfand, and Jona Lasinio (2012). Thus, only central concepts of Gaussian processes are introduced here and the interested reader is pointed to the above resources for more details.

Rasmussen and Williams (2006) point out incisively that a Gaussian process can ultimately be seen as "a generalization of the Gaussian probability distribution" (p. 2) as it provides a way to extend the multivariate Gaussian distribution to the framework of stochastic processes and as such defines a distribution over functions. While this sounds incredibly powerful, it might also be an unintuitive statement for readers unfamiliar with Gaussian processes and thereby calls for clarification and a closer look on the properties of GPs and how they can be used as a tool in modeling and prediction.

2.1.1 Motivation: Searching in a Space of Functions

In general, to gain knowledge from data one needs to explore meaningful connections between a given set of input values or explanatory variables and outputs or dependent variables. As Rasmussen and Williams (2006) argue, it is a general goal of modeling efforts to search for predictions of the output values that correspond to new and previously unobserved inputs, i.e. for a dataset $\mathcal{D} = \{(\mathbf{x_i}, y_i)|i = 1, ..., n\}$ of n observations with input vector $\mathbf{x}$ and outputs y, the objective is to find predictions for output values corresponding to unobserved inputs denoted $\mathbf{x}_\star$.
Loosely speaking, to arrive at such predictions one aims to find a formal rule (i.e. a function) describing the connections between input variables and output values using the known data available in $\mathcal{D}$. This process is often called "training", especially in machine learning setups, and as such $\mathcal{D}$ is also often referred to as training data (cf. Rasmussen and Williams (2006)). As Rasmussen and Williams (2006) point out concisely, the set of such possible formal rules has to be actively limited "as otherwise any function which is consistent with the training data would be equally valid" (Rasmussen and Williams (2006), p.2).

For a parametric approach like a linear regression model, outputs and inputs are assumed to be connected by $y_k = x_{k1}\beta_1 + ... + x_{kn}\beta_n + \varepsilon_k$ for $k = 1, ..., n$. Thus, the space of possible functions is already limited by assuming linear functions with a certain parametrization $\boldsymbol{\beta} = \begin{bmatrix}\beta_1 & \beta_2 & ... & \beta_n\end{bmatrix}^T$.

But what can be done if one wants to search directly in the space of possible functions without limiting the scope to a certain type of function and parametrization beforehand? Instead of finding estimates for the parameters of a model, one could imagine to directly search in the space of all possible functions with the goal to find a distribution over possible functions and ultimately the most "probable" functions for describing a given problem (cf. Murphy (2012), p. 518). It will be illustrated that Gaussian processes can in fact enable such a search in finite time, even though the space of all possible functions is infinite, by offering the possibility to control properties like the general shape or periodicity of such functions (see Rasmussen and Williams (2006)).

2.1.2 Multivariate Gaussian

To build towards an understanding of Gaussian processes, the next subsection follows the intuitive approach in Görtler, Kehlbeck, and Deussen (2019) by introducing and revisiting the multivariate Gaussian distribution and some of its properties. This will lateron be useful for shifting the focus from estimating parameter values to searching a space of different possible functions with different parametrizations that could describe connections between given inputs and outputs.

Two random variables Y_1 and Y_2 are jointly Gaussian distributed when $\mathbf{Y} = \binom{Y_1}{Y_2} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The expected value of such a multivariate Gaussian distribution is characterized by the mean vector $\boldsymbol{\mu} = \binom{\mu_1}{\mu_2}$ and the shape of the distribution is described by the covariance matrix $\boldsymbol{\Sigma} = \binom{\Sigma_{11}\ \Sigma_{12}}{\Sigma_{21}\ \Sigma_{22}}$. These two components completely identify a given multivariate Gaussian distribution (cf. Härdle and Simar (2019)).

Multivariate Gaussian distributions have two particularly useful properties that will be revisited briefly. Most notably, if a joint Gaussian distribution of two random variables Y_1 and Y_2 is given, the conditional distribution $Y_1|Y_2 \leq y_2 \sim \mathcal{N}(\bar{\mu}, \bar{\Sigma})$ is directly accessible with $\bar{\mu} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(y_2 - \mu_2)$ and $\bar{\Sigma} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$ as it is shown in e.g. Held and Sabanés Bové (2014) and Härdle and Simar (2019). This means that if the joint distribution of a vector of Gaussian distributed random variables is known, it is possible to obtain marginal and conditional distributions that will also be Gaussian, i.e. the Gaussian distribution is closed under marginalization and conditioning. Thereby, marginalization allows to extract partial information from a joint Gaussian distribution and conditioning can be used to determine conditional probabilities that again follow a Gaussian distribution. In what follows, these

properties of multivariate Gaussian distributions facilitate formulating prior and posterior distributions and performing Bayesian inference with Gaussian processes.

2.1.3 From a Multivariate Gaussian to a Gaussian Process

For a two-dimensional Gaussian distribution as introduced above, a contour plot as in Figure 2 can illustrate the mean vector and the covariance matrix. Shi (2019) insightfully illustrates the process of sampling different points from multivariate Gaussians with animated plots and builds up towards an intuition of Gaussian processes summarizing an introductory tutorial by Turner (2016). Following the ideas and visualizations presented therein and in Turner (2016) as well as in Klein (2019), some plots are reproduced here to foster an intuitive understanding of the construction of functions from multivariate Gaussian distributions.

As a starting point, one can imagine sampling points from a two-dimensional Gaussian. Each sample consists of two coordinates (y_1, y_2) that denote the realizations of Y_1 and Y_2 at the sampled point. Now, these coordinates of the samples are transferred to another coordinate system where the abscissa consists of the indices of the variables and the ordinate measures the value of y_1 and y_2 at the selected sample point.
The sampled values y_1 and y_2 are closer, the higher Y_1 and Y_2 are correlated and vice versa. Thus, the covariance matrix of Y_1 and Y_2 informs about the extent of differences in the sampled results.

Using samples from a bivariate Gaussian distribution with mean zero and covariance 0.8, Figure 2 illustrates this transfer step with two samples from a bivariate Gaussian distribution depicted as dots in Figure 2 (A) that are then plotted in Figure 2 (B).
The key step is to shift the perspective such that the indices of y_1 and y_2 can be thought of as inputs to a function $f(.)$ (cf. Klein (2019)). The function values $f(1)$ and $f(2)$ of that function $f(.)$ are the values of y_1 and y_2 for the given sample, i.e. $f(1) = y_1$ and $f(2) = y_2$. Each sample thereby creates a realization of function values of a function $f(.)$ that is defined on the domain of the indices as illustrated in Figure 2 (B).

From this point of view, conditioning on y_1 comes down to fixing y_1 to a specific value and sampling y_2 conditional on y_1. This procedure is illustrated in Figure 3 and leads to a fixed starting point of the line in Figure 3 (B) while the second point varies between the different samples. Intuitively, extending this setup to a multivariate Gaussian in higher dimensions leads to more indices, i.e. more variables on the abscissa. In Figure 4, results for a three-

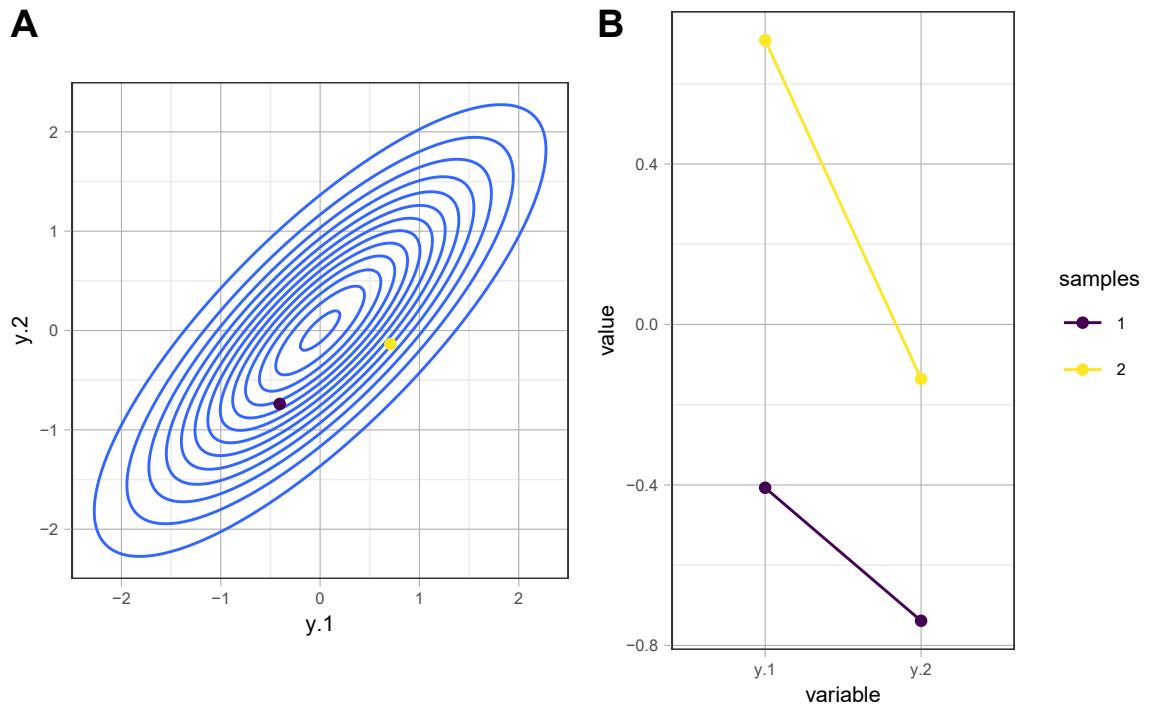


Figure 2: (A) shows a contour plot for a bivariate Gaussian distribution with mean zero and covariance 0.8 created with `dmvnorm()` in the `R`-package `mvtnorm`. (B) shows the transfer step. Plots are produced by the author following Turner (2016), Shi (2019) and Klein (2019).

dimensional multivariate Gaussian are presented assuming the covariance matrix:

$$\begin{bmatrix} 1 & 0.8 & 0.4 \\ 0.8 & 1 & 0.7 \\ 0.4 & 0.7 & 1 \end{bmatrix}$$

Figure 4 (A) depicts conditioning on y_1 and drawing ten samples for y_2 and y_3 conditioned on y_1. Figure 4 (B) shows another sample from the three-dimensional Gaussian conditioned on a specific value for y_1 and y_3.

Sampling from higher-dimensional Gaussian distributions eventually leads to a finer grid of values on the abscissa. The higher the dimensions of the multivariate Gaussian, the more these rearranged sampled points start to resemble a function.

One can repeat the sampling of different possible functions given a high-dimensional Gaussian distribution. This creates a set of functions that can be conditioned on given observations. A mean and a variance for each sample of possible curves could be calculated, thus allowing to assess the sampled functions (cf. Shi (2019)).

At this point, it becomes possible to directly sample functions conditioned on observed data points. Still, the abscissa consists of indices (i.e. integers).

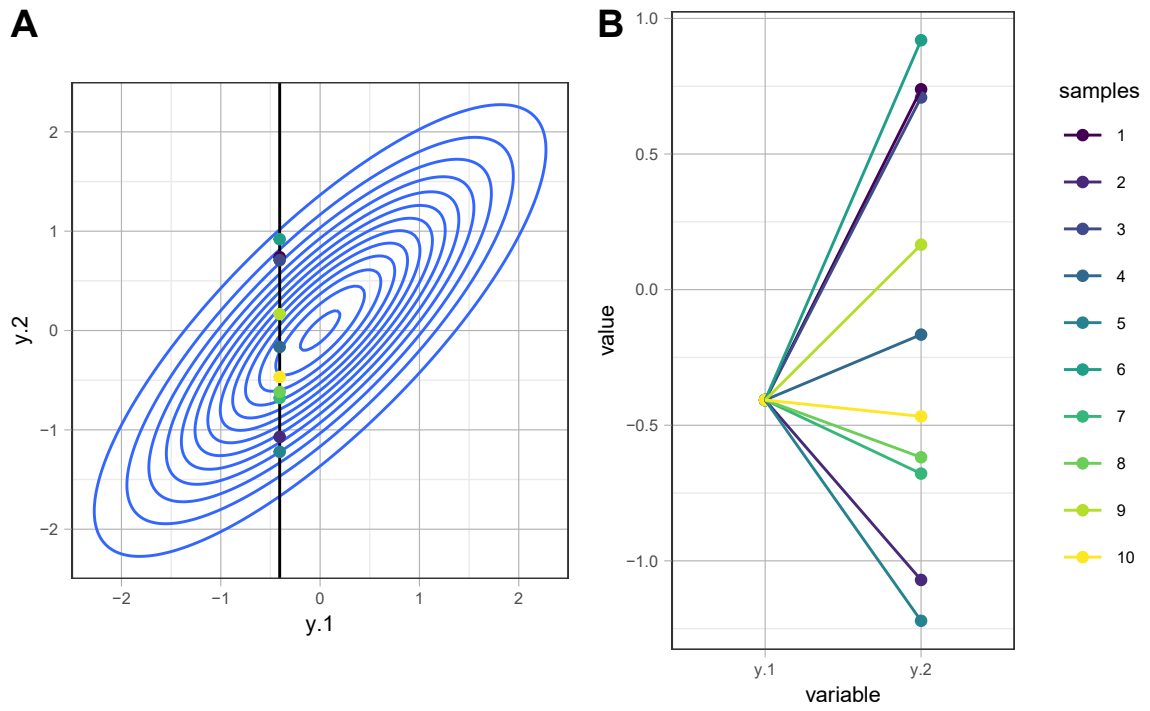


Figure 3: (A) shows a contour plot for a bivariate Gaussian distribution with mean zero and covariance 0.8 conditioned on $y_1 = -0.4070352$. (B) shows the transfer step. Plots are produced by the author following Turner (2016), Shi (2019) and Klein (2019).

As pointed out in Klein (2019), generalizing the results to an infinite domain is not possible by merely increasing the dimensionality of the multivariate Gaussian to achieve a finer grid of points. For this, one can make use of kernel functions (cf. Turner (2016) and Shi (2019)).

2.1.4 Kernel functions

In general, kernel functions $\kappa(.,.)$ denote functions that map two arguments from a given domain of input data $\mathcal{X}$ to $\mathbb{R}$, i.e. $\kappa(\mathbf{x}, \mathbf{x}') \in \mathbb{R}$ for $\mathbf{x}, \mathbf{x}' \in \mathcal{D}$ (see Rasmussen and Williams (2006) and Murphy (2012)). A kernel function is symmetric if $\kappa(\mathbf{x}, \mathbf{x}') = \kappa(\mathbf{x}', \mathbf{x})$ and non-negative if $\kappa(\mathbf{x}, \mathbf{x}') \geq 0$ (see Murphy (2012)). It can be used to generate a so-called Gram matrix $G_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ for a given input dataset $\{\mathbf{x}_i | i = 1, ..., n\}$ that captures relations between all input points. If additionally, the kernel function is symmetric and non-negative, the matrix G is a covariance matrix and $\kappa(.,.)$ is a covariance function (see Rasmussen and Williams (2006)).

Making use of these properties, such kernel functions can provide a measure of similarity of input values in a given dataset and serve as covariance functions generating a covariance

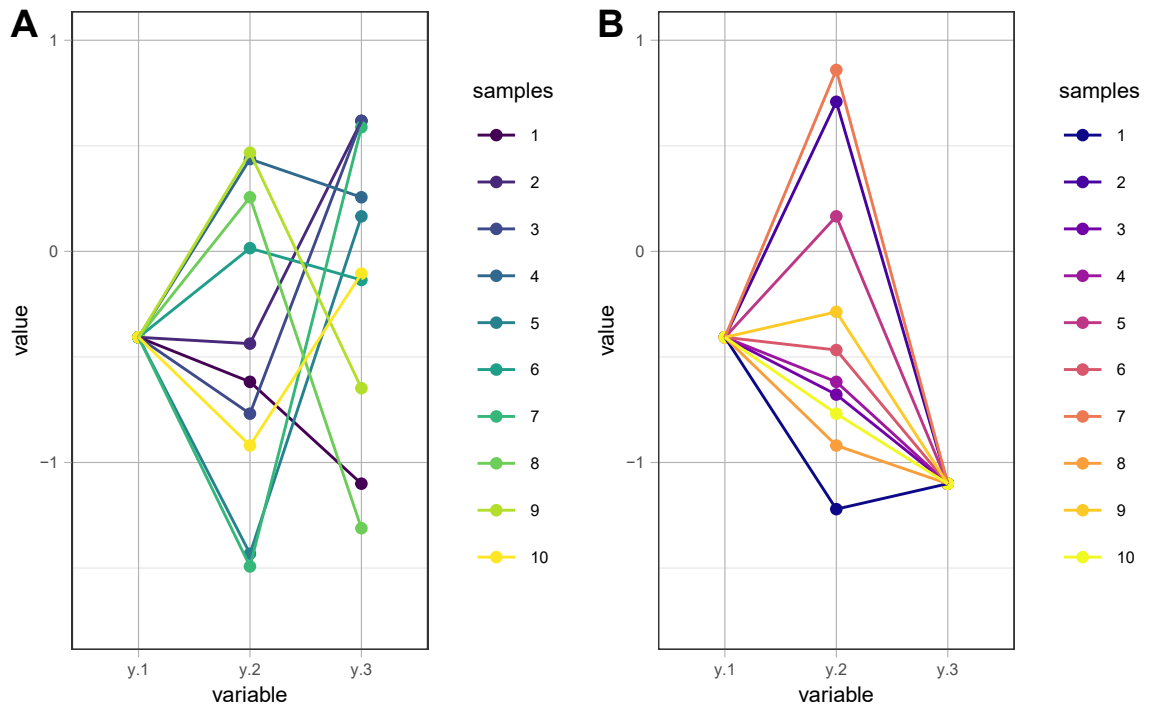


Figure 4: Two results of sampling from a three-dimensional Gaussian distribution. (A) shows results conditioned on y_1, (B) conditioned on y_1 and y_3. Plots are produced by the author following Turner (2016) and Shi (2019).

matrix from real-valued observations, thus overcoming the limitations to an integer-valued domain as illustrated and visualized in great detail in Turner (2016), Shi (2019) and Görtler, Kehlbeck, and Deussen (2019). Kernel functions that are covariance functions offer a formal rule for the creation of a covariance matrix, thereby enabling the definition of a quasi-infinite-dimensional Gaussian distribution that can ultimately be conceptualized as a Gaussian process (cf. Turner (2016) and Shi (2019)). It becomes apparent now that in fact, as pointed out earlier, Gaussian processes enable a generalization of the multivariate Gaussian distribution to an infinite domain (see Rasmussen and Williams (2006)).

As desired, samples from this Gaussian process are smooth functions with infinite domain size, i.e. one can define a distribution over functions with a Gaussian process and obtain samples of possible functions from this distribution over functions.

2.1.5 Definition of a GP

To summarize the building blocks introduced above, a general definition of Gaussian processes is provided here. Formally, one can write down a Gaussian process as a distribution over functions $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'))$, i.e. $f(\mathbf{x})$ follows a distribution defined by a Gaussian process with a mean function $m(\mathbf{x})$ and covariance function $\kappa(\mathbf{x}, \mathbf{x}')$ (see Rasmussen and

Williams (2006) and Murphy (2012)). As introduced earlier, an indexed collection of random variables with a dependence structure over an index set is called Gaussian process when any finite subset of these random variables is jointly Gaussian distributed (cf. Rasmussen and Williams (2006)).
A Gaussian process is completely defined by the mean function $m(\mathbf{x})$ (also denoted $\mu(\mathbf{x})$) and the covariance function $\kappa(\mathbf{x}, \mathbf{x}')$ for given inputs $\mathbf{x}$ and $\mathbf{x}'$ (see Rasmussen and Williams (2006)), in the same sense as a multivariate Gaussian distribution is fully characterized by its mean vector and covariance matrix.

2.2 Predicting with a GP

The next parts show in more detail how to think of a Gaussian process as a distribution over functions and how this enables to perform inference and prediction in such a space of functions in a Bayesian framework.

2.2.1 Gaussian processes in a Bayesian framework

Bayesian inference is a widely applied approach in different fields of research. The focus of this section will not be on Bayesian statistics in general, see Held and Sabanés Bové (2014) or Kruschke (2015) for a general introduction to Bayesian statistics in contrast to frequentist approaches. Also, Diggle and Ribeiro jr. (2007) provide an in-depth introduction of Bayesian inference in geostatistics. Therein, the authors introduce the Bayesian framework as a general and widely used approach to account for uncertainty in the estimation of model parameters and to establish a joint framework of parameter estimation and predictive inference.

A core element of Bayesian inference is the notion of modeling and updating beliefs after observing data by making use of relationships between joint, marginal and conditional probability distributions. These relations are formalized by Bayes Theorem which states that $h(y_2|y_1) = \frac{h(y_1,y_2)}{h(y_1)} = \frac{h(y_1|y_2)h(y_2)}{h(y_1)}$ for two continuous random variables Y_1 and Y_2 with realizations y_1, y_2 and a joint probability density function $h(y_1, y_2)$ such that $\mathbb{P}[Y_1 \leq y_1, Y_2 \leq y_2] = \int_{-\infty}^{y_2} \int_{-\infty}^{y_1} h(u, v) du\ dv$. This relation follows from the definition of conditional probabilities (see also Held and Sabanés Bové (2014), p. 318).
In contrast to frequentist statistics, prior knowledge and beliefs are explicitly modeled in Bayesian approaches by assuming probability distributions over the unknown model parameters (also known as prior distributions). A posterior distribution is then established as an updated version of the prior distribution after observing data by applying Bayes Theorem.

In particular, connections of prior distribution, likelihood and marginal likelihood of a given model to the posterior distribution given a set of parameters and input data are summarized by the relation: posterior $= \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$ (see Rasmussen and Williams (2006) and Held and Sabanés Bové (2014), p.170). Inferential insights about the model parameters rest upon this posterior distribution that combines prior beliefs and observed data, e.g. point estimates for the unknown model parameters can be obtained as expected values of this posterior distribution (see Held and Sabanés Bové (2014), p.171).

As established earlier, Gaussian processes can define a probability distribution over functions. Again, a Gaussian process is completely described by a mean function $m(\mathbf{x})$ and the covariance (or kernel) function $\kappa(\mathbf{x}, \mathbf{x'})$ that defines covariance matrices. By selecting specific mean and covariance functions, Gaussian processes can be used to establish a prior distribution over possible functions.
Murphy (2012) points out that often the mean function is set to zero as Gaussian processes are able "to model the mean arbitrarily well" (Murphy (2012), p. 518). At the same time, the covariance function provides a generating rule for the covariance matrices thereby playing an integral role in shaping the functions generated by the given Gaussian process. Selecting kernel functions and choosing values for the parameters in a given kernel function are therefore the essential steps for including prior beliefs and assumptions in a Gaussian process model (cf. Rasmussen and Williams (2006), p.79).

Visualizations and animations in Görtler, Kehlbeck, and Deussen (2019) illustrate interactively how the creation of the covariance matrix with a given kernel function enables incorporating prior assumptions and thereby influences which types of functions are more probable, e.g. a linear kernel can be used for creating different linear functions or a periodic kernel for sampling functions with assumed periodicity.
One can think of the kernel function of a GP controlling prior specifications in two steps. First, the selected kernel function limits the class or type of functions that are considered. Secondly, by changing the parameters of the chosen kernel function (also called hyperparameters), properties like the smoothness or the variation in the functions that are sampled from the Gaussian process can additionally be controlled (see Rasmussen and Williams (2006), p. 20).
Thereby, prior specifications of a Gaussian process set boundaries to e.g. shapes, smoothness, concentration around the mean or periodic patterns of the considered functions (cf.

Rasmussen and Williams (2006) and Görtler, Kehlbeck, and Deussen (2019)). In that way, GPs define a prior over possible functions and, given newly observed data, ultimately allow to obtain a posterior distribution over functions. This procedure of posterior prediction in a space of functions will be explained in more detail in the next part.

2.2.2 Predicting in a Space of Functions

Following introductory examples in Rasmussen and Williams (2006), imagine a given dataset with inputs and outputs given as $\{\mathbf{x}_i, f_i | i = 1, ..., n\}$ where $f_i = f(\mathbf{x}_i)$ and $\mathbf{f} = (f_1, ...f_n)^T$. Observations are assumed to be noise-free for simplicity. Also assume that there is new test data available with a matrix of new inputs $\mathbf{X}_\star$ and an unknown corresponding vector of outputs $\mathbf{f}_\star$.

To get an estimate of $\mathbf{f}_\star$, one can use the defining properties of a Gaussian process and assume that the test data stems from the same distribution as the training data, i.e. the vector of given outputs denoted $\mathbf{f}$ and $\mathbf{f}_\star$ are jointly Gaussian distributed. The joint distribution of the training outputs $\mathbf{f}$ and the newly observed outputs $\mathbf{f}_\star$ is given by:

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{f}_\star \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_\star \end{pmatrix}, \begin{pmatrix} \mathbf{S} & \mathbf{S}_\star \\ \mathbf{S}_\star^T & \mathbf{S}_{\star\star} \end{pmatrix} \right) \tag{1}$$

Here, $\mathbf{S}$ is the covariance matrix of the training data $\mathbf{X}$ created via a selected kernel function $\kappa(.,.)$. $\mathbf{S}_{\star\star}$ is the variance of the test value as calculated via the same kernel function and $\mathbf{S}_\star$ is the covariance between the training and test data points. It is important to note that the dimensions of the multivariate Gaussian distribution change when adding additional data.

To get the conditional distribution of $\mathbf{f}_\star$ given $\mathbf{f}$ and the training and newly observed test data points, one can make use of the rules for conditional Gaussian distributions above introduced. This leads to mean $\boldsymbol{\mu}_\star = \boldsymbol{\mu}(\mathbf{X}_\star) + \mathbf{S}_\star^T \mathbf{S}^{-1}(\mathbf{f} - \boldsymbol{\mu}(\mathbf{X}))$ and covariance matrix $\boldsymbol{\Sigma}_\star = \mathbf{S}_{\star\star} - \mathbf{S}_\star^T \mathbf{S}^{-1} \mathbf{S}_\star$ for the conditional distribution of the newly observed data given the training observations, i.e. $(\mathbf{f}_\star | \mathbf{X}_\star, \mathbf{X}, \mathbf{f}) \sim \mathcal{N}(\boldsymbol{\mu}_\star, \boldsymbol{\Sigma}_\star)$.

For a setup with assumed zero mean, $\boldsymbol{\mu}_\star = \boldsymbol{\mu}(\mathbf{X}_\star) + \mathbf{S}_\star^T \mathbf{S}^{-1}(\mathbf{f} - \boldsymbol{\mu}(\mathbf{X}))$ reduces to $\boldsymbol{\mu}_\star = \mathbf{S}_\star^T \mathbf{S}^{-1} \mathbf{f}$ (see Rasmussen and Williams (2006), p. 16).

In other words, one can obtain a posterior distribution over functions $\mathbf{f}_\star$ given $\mathbf{f}$ as well as the training data inputs $\mathbf{X}$ and newly observed input data $\mathbf{X}_\star$.

Function values for the newly observed inputs can be sampled from this joint posterior distribution making use of the mean and covariance matrix obtained from the above results. Ras-

mussen and Williams (2006) provide a detailed explanation of the steps that allow to generate such samples with scalar Gaussian generators in most programmming environments making use of Cholesky decomposition and standard Gaussian vectors (see Rasmussen and Williams (2006), p.200). Figure 5 illustrates the procedure for the noise-free example with zero mean and a squared exponential kernel function $\mathbf{Cov}[f(\mathbf{x}_i), f(\mathbf{x}_j)] = \kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{exp}(-\frac{1}{2}|\mathbf{x}_i - \mathbf{x}_j|^2)$. The plots in Figure 5 reproduce results presented in Rasmussen and Williams (2006) p.15 and Murphy (2012) p.519 with the help of insightful R-code by Clark (2020). Figure 5 (A)

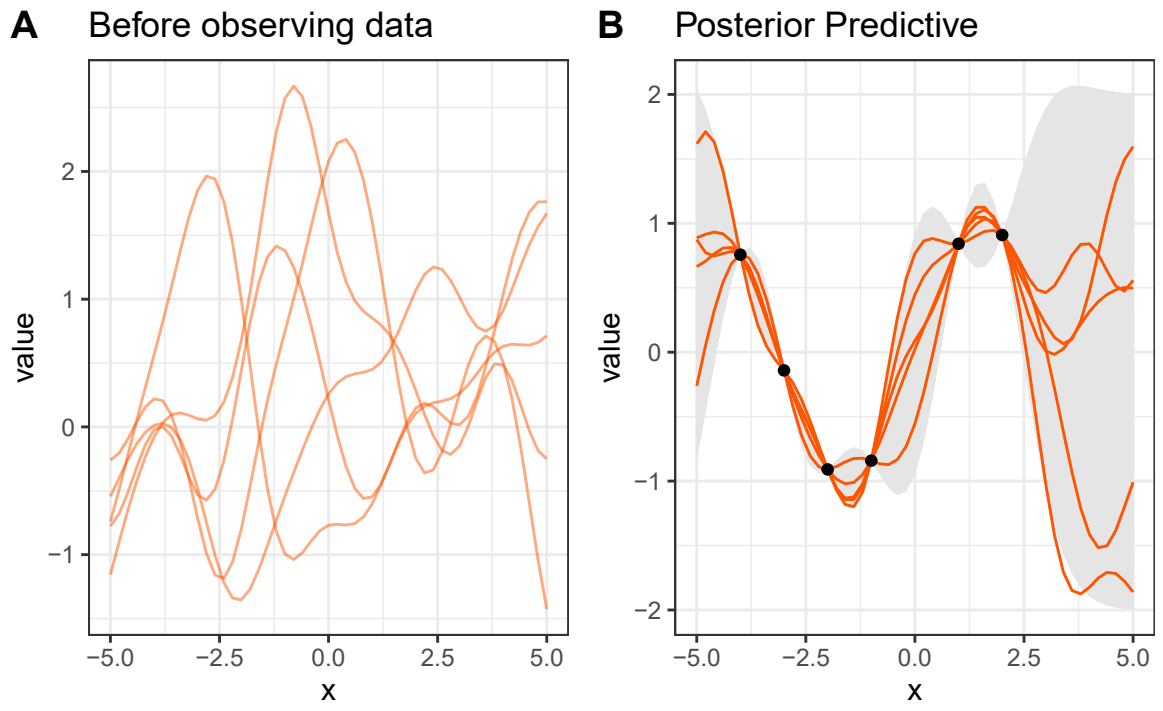


Figure 5: (A) shows five functions drawn from GP prior with mean=0 and squared exponential covariance function, joining a large number of points allows to depict functions as lines. (B) shows five functions drawn from the posterior (i.e. conditioned on the six observations marked by black dots). The shaded areas correspond to a 95%-confidence region indicating low confidence where there are no observations available. These plots reproduce results presented in Rasmussen and Williams (2006) and Murphy (2012) with the help of R-code by Clark (2020).

shows five functions drawn from the GP prior with zero mean and squared exponential covariance function. By sampling functions from the defined Gaussian process conditioned on a set of observed data points depicted as black dots in Figure 5 (B), one automatically gets a confidence band for the estimated functions. Variances are higher in areas where no data is available which makes intuitive sense as uncertainty should be higher when less data is available.

In that way, the posterior predictive obtained with a Gaussian process prior over functions includes immediate modeling of uncertainty for posterior sampled functions. This a ma-

jor advantage of Gaussian process models compared to parametric modeling approaches, e.g. non-linear regression, where uncertainty can only be assessed for the estimates of the model parameters of an explicitly selected parametrization. Continuing the comparison, Gaussian process models are in fact non-parametric models, i.e. they do not require selecting a specific set of parameters but instead generate sets of functions (cf. Rasmussen and Williams (2006)).

2.3 Gaussian Processes and Spatial Circular Data

Building on the intuitions for Gaussian processes and how they can be used for predicting functions, the next part shows how the framework of Gaussian processes can be extended to the analysis of circular spatial data.
As introduced earlier, circular data has unique characteristics that motivate circular modeling approaches. The von-Mises distribution is one of the earliest and most well-known approaches for defining a distribution on a circular domain in the univariate case (cf. Mardia and Jupp (2000), p. 36). Broadly speaking, the main idea of the von-Mises distribution is to define a probability density function that is periodic over the domain of the circular variable, i.e. over $[0, 2\pi)$ or, without loss of generality, $[-\pi, \pi)$ (see Coles (1998) for more details).

This thesis focuses on circular data measured at different locations in space, i.e. circular spatial measurements. Even though the circular von-Mises distribution and its proximity to the Gaussian distribution is useful in some setups, there are more general ways of modeling circular data, e.g. wrapping densities around the unit circle (cf. Coles (1998)). Moreover, Jona Lasinio, Gelfand, and Jona Lasinio (2012) note that using the von-Mises distribution in bivariate or trivariate settings gets complex and computationally expensive very fast. As such, they conclude that (for now) an approach based on the von-Mises distribution is not feasible for high-dimensional settings like the analysis of spatial observations.

Referring to the time when writing their paper, Jona Lasinio, Gelfand, and Jona Lasinio (2012) notice that "[t]here is little in the way of formal multivariate theory of circular data, particularly in the fully Bayesian setting" (Jona Lasinio, Gelfand, and Jona Lasinio (2012), p. 2). The work of Coles (1998) is mentioned as an early example for the analysis of wind directions over time using a wrapping approach in a Bayesian framework. Jona Lasinio, Gelfand, and Jona Lasinio (2012) develop a wrapping approach following Coles (1998) and Coles and Casson (1998) and introduce wrapped spatial Gaussian process models in a Bayesian hierar-

chical modeling framework. Extending ideas in Coles (1998) and Coles and Casson (1998) to a spatial setting, Jona Lasinio, Gelfand, and Jona Lasinio (2012) transform linear Gaussian processes to a circular spatial setup that allows to directly model spatial dependence of the circular variables.

2.3.1 Gaussian distributions for directional data

This part will firstly present how wrapping transfers a regular Gaussian distribution to a distribution that accounts for the circular characteristics of the given variable. Subsequently, these results are generalized to the realm of stochastic processes, focusing on the spatial wrapped Gaussian process model from Jona Lasinio, Gelfand, and Jona Lasinio (2012).[1]

Wrapped Gaussian distribution A wrapped distribution is created by wrapping a probability density defined on $\mathbb{R}$ around the unit circle to obtain a circular function. For a linear random variable $Y \in \mathbb{R}$, the wrapped version of Y of period 2π is given by $\Theta = Y \mod 2\pi$ with $\Theta \in [0, 2\pi)$ (see Mardia and Jupp (2000) and Jona Lasinio, Gelfand, and Jona Lasinio (2012)). In that way, there exists a one-to-one relationship between a linear and a circular variable in the wrapping approach.

Assuming that the random variable Y has a probability density function $h(y)$, this function $h(y)$ can be wrapped around the unit circle (see Coles (1998) and Jona Lasinio, Gelfand, and Jona Lasinio (2012)). The relation $\Theta = Y \mod 2\pi$ can be rewritten as $\Theta = Y - 2\pi K$ using a winding number $K \in \mathbb{Z} \equiv \{0, \pm 1, \pm 2, ...\}$. The probability density function of the induced circular variable Θ with realizations θ is then $g(\theta) = \sum_{k=-\infty}^{\infty} h(\theta + 2\pi k)$ (cf. Mastrantonio, Jona Lasinio, and Gelfand (2016)). This is possible because $h(y)$ is a probability density function and one can think of mod 2π as transforming a linear variable by wrapping it around the circle where a higher value of Y corresponds to more rounds of wrapping around the circle (i.e. a larger winding number) to turn it into a circular variable.

For $Y \sim \mathcal{N}(\mu, \sigma^2)$, the probability density $h(y)$ is the Gaussian distribution with mean μ and variance σ^2. Accordingly, the corresponding circular variable Θ follows a wrapped Gaussian

[1] It is worthwhile to mention that there are of course other ways to formulate distributions and models for circular data. For example, there are approaches working with projections where the main idea is to picture two random variables Y_1 and Y_2 as coordinates in an Euclidean space and make use of the relation $\tan(\Theta) = \frac{Y_1}{Y_2}$ to relate Y_1 and Y_2 to the angle Θ between the vectors spanned by the random variables. Then, one can use a variant of the inverse tangent function to get the circular variable Θ (see Mastrantonio, Jona Lasinio, and Gelfand (2016) or Jona Lasinio, Santoro, and Mastrantonio (2020) for more details)

distribution $\mathcal{WN}(\mu, c)$ and has the probability density function:

$$g(\theta; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} \exp\left(\frac{-(\theta - \mu + 2\pi k)^2}{2\sigma^2}\right) \tag{2}$$

where σ^2 can be reparametrized using the concentration parameter $c = e^{-\sigma^2/2} < 1$ and $0 \leq \theta < 2\pi$ (see Mardia and Jupp (2000) and Jona Lasinio, Gelfand, and Jona Lasinio (2012), p. 5).

Thereby, wrapping can create circular distributions similar to their linear counterparts e.g. the wrapped Gaussian distribution is still unimodal and symmetric (see chapter 3 in Mardia and Jupp (2000) for other useful properties of the wrapped Gaussian distribution).

Introducing a Latent Winding Number The crucial step for enabling model fitting and inference in the above setup is concisely introduced in Coles (1998) as "treating the wrapping coefficients as missing data" (Coles (1998), p. 106). Therein, this allows the author to use Markov chain Monte Carlo (MCMC) methods to develop a framework for inference in a wrapping setup. As pointed out by Jona Lasinio, Gelfand, and Jona Lasinio (2012), introducing winding numbers as latent variables means sampling winding numbers at each location and iteration which is challenging as $K \in \mathbb{Z} \equiv \{0, \pm 1, \pm 2, ...\}$.

In spite of that, by reformulating the univariate wrapped Gaussian distribution $g(\theta; \mu, \sigma^2)$, Mardia and Jupp (2000) show that an approximation of the wrapped Gaussian requires only a small set of values k for K. Using properties of wrapped distributions and the characteristic function of the wrapped Gaussian distribution, they conclude that in practice the wrapped Gaussian density can be approximated with $k \in \{-1, 0, 1\}$ for $\sigma^2 \geq 2\pi$ and $k = 0$ for $\sigma^2 < 2\pi$. For a detailed derivation of these results see Mardia and Jupp (2000) p.49f.

In a similar spirit, Jona Lasinio, Gelfand, and Jona Lasinio (2012) extend the result in Mardia and Jupp (2000) providing more precise criteria for the selection of winding numbers. In particular, they set $\Theta' = (\Theta + \pi) \bmod 2\pi - \pi$ to get a symmetric support for θ, i.e. $-\pi \leq \theta \leq \pi$ and rewrite $g(\theta'; \mu', \sigma^2)$ as $\frac{1}{\sigma}\varphi\left(\frac{\theta' + 2\pi k - \mu'}{\sigma}\right)$ where $\varphi(.)$ denotes the standard Normal Gaussian density function.

As derived in Jona Lasinio, Gelfand, and Jona Lasinio (2012) and further explained in G. Jona Lasinio, Gelfand, and Jona Lasinio (2019), these results enable a selection of the number of winding numbers needed for approximating the wrapped density by using the variance σ^2 e.g. for $\sigma < 2\pi/3$, the authors derive that $k \in \{-1, 0, 1\}$ (cf. Jona Lasinio, Gelfand, and Jona Lasinio (2012), p.6 and G. Jona Lasinio, Gelfand, and Jona Lasinio (2019), p. 132). For more

details on these results consider Jona Lasinio, Gelfand, and Jona Lasinio (2012) p.5f.

Multivariate Setup Emphasized by Coles (1998) as a great advantage of wrapping, such modeling setups can be generalized to circular multivariate variables as the wrapping transformation can as well be applied to each component of a multidimensional variable. For a vector $\mathbf{Y} = (Y_1, ..., Y_p)^T$ of linear random variables and a vector $\mathbf{\Theta} = (\Theta_1, ..., \Theta_p)^T$ of circular variables, one can write $\Theta_i = Y_i \mod 2\pi$ where $i = 1, ..., p$. When $\mathbf{Y}$ has a p-variate distribution $h(.)$ one can again make use of the transformation $\mathbf{\Theta} = \mathbf{Y} + 2\pi\mathbf{K}$ to get to a distribution of $\mathbf{\Theta}$. As shown in G. Jona Lasinio, Gelfand, and Jona Lasinio (2019), this wrapped distribution is given by:

$$g(\boldsymbol{\theta}) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} ... \sum_{k_p=-\infty}^{\infty} h(\boldsymbol{\theta} + 2\pi\mathbf{k}) \tag{3}$$

From the above equation, one can see that $g(\boldsymbol{\theta})$ is given by summing up $g(\boldsymbol{\theta}, \mathbf{k})$ over all k_i with $i = 1, ..., p$. Therefore, the joint distribution of $\mathbf{\Theta}$ and $\mathbf{K}$ is $g(\boldsymbol{\theta}, \mathbf{k}) = h(\boldsymbol{\theta} + 2\pi\mathbf{k})$ (cf. Jona Lasinio, Gelfand, and Jona Lasinio (2012)).
For a p-variate Gaussian density function $h(.; \boldsymbol{\eta})$ with parameters $\boldsymbol{\eta} = (\boldsymbol{\mu}, \mathbf{\Sigma})$, the induced circular variable $\mathbf{\Theta}$ therefore follows a p-variate wrapped Gaussian distribution with parameters $(\boldsymbol{\mu}, \mathbf{\Sigma})$.

Starting from the ideas for a latent winding number presented in section 1, Jona Lasinio, Gelfand, and Jona Lasinio (2012) derive the marginal and conditional distribution of the vector of winding numbers and show how these results can be used when introducing a latent random vector $\mathbf{K}$ of winding numbers. In particular, following Jona Lasinio, Gelfand, and Jona Lasinio (2012) and the results presented above, one can formulate the following expressions for the conditional probabilities $\mathbb{P}[\mathbf{K} = \mathbf{k}|\mathbf{\Theta} = \boldsymbol{\theta}]$ and $\mathbb{P}[\mathbf{\Theta} = \boldsymbol{\theta}|\mathbf{K} = \mathbf{k}]$ for the multivariate setting:

$$\mathbb{P}[\mathbf{K} = \mathbf{k}|\mathbf{\Theta} = \boldsymbol{\theta}] = \frac{\mathbb{P}[\mathbf{\Theta} = \boldsymbol{\theta}, \mathbf{K} = \mathbf{k}]}{\mathbb{P}[\mathbf{\Theta} = \boldsymbol{\theta}]} = \frac{h(\boldsymbol{\theta} + 2\pi\mathbf{k})}{\sum_{j_1=-\infty}^{\infty} ... \sum_{j_p=-\infty}^{\infty} h(\boldsymbol{\theta} + 2\pi\mathbf{j})} \tag{4}$$

$$\mathbb{P}[\mathbf{\Theta} = \boldsymbol{\theta}|\mathbf{K} = \mathbf{k}] = \frac{\mathbb{P}[\mathbf{\Theta} = \boldsymbol{\theta}, \mathbf{K} = \mathbf{k}]}{\mathbb{P}[\mathbf{K} = \mathbf{k}]} = \frac{h(\boldsymbol{\theta} + 2\pi\mathbf{k})}{\int_0^{2\pi} ... \int_0^{2\pi} h(\boldsymbol{\theta} + 2\pi\mathbf{k}) d\boldsymbol{\theta}} \tag{5}$$

Thus, the vector of winding numbers $\mathbf{K}$ with realizations $\mathbf{k}$ can be treated as a latent variable and truncation results illustrated in section 1 again limit this sampling of winding numbers (cf. Jona Lasinio, Gelfand, and Jona Lasinio (2012), p.4).

2.3.2 Wrapped Spatial Gaussian processes

Jona Lasinio, Santoro, and Mastrantonio (2020) introduce a general p-variate spatial Gaussian process defined over a d-dimensional domain as $Y(\mathbf{s}) = \{Y_i(\mathbf{s})_{i=1}^p \in \mathbb{R}^p\}$ where $\mathbf{s} \in \mathcal{S} \subset \mathbb{R}^d$ denote locations in a d-dimensional space. Following the results presented for spatial stochastic processes by Gneiting and Guttorp (2010), such a spatial stochastic process has realizations $\mathbf{Y} = (Y_1, Y_2, ..., Y_n)^T = (Y(\mathbf{s}_1), Y(\mathbf{s}_2), ..., Y(\mathbf{s}_n))^T \in \mathbb{R}^p \times \mathbb{R}^n$ that follow a collection of joint finite-dimensional distributions $H\left(Y(\mathbf{s}_1), ..., Y(\mathbf{s}_n); \mathbf{s}_1, ..., \mathbf{s}_n\right)$. In the case of a Gaussian process, realizations in fact follow joint multivariate Gaussian distributions per definition of a GP (cf. section 2.1.5). For $p = 1$, the above expressions describe a wrapped spatial GP which will be considered in the following[2].

Reflecting the correspondence of a wrapped multivariate Gaussian distribution to a linear multivariate Gaussian distribution introduced above, there is a direct connection between a linear Gaussian process and its wrapped counterpart: "A Gaussian process on $\mathbb{R}^d$ induces a wrapped Gaussian process on $\mathbb{R}^{d}$" (Jona Lasinio, Gelfand, and Jona Lasinio (2012), p. 8). Thereby, the wrapping approach extends further to the realm of stochastic processes by setting $\Theta(\mathbf{s}) = Y(\mathbf{s}) \bmod 2\pi$.

Following the notation in G. Jona Lasinio, Gelfand, and Jona Lasinio (2019), let a set of spatial locations be given by $\{\mathbf{s}_i : i = 1, ..., n\}$ where each entry $\mathbf{s}_i \in \mathcal{S} \subset \mathbb{R}^d$ with $d = 2$, as only spatial distances h_{sp} are considered here.[3] Then, a wrapped spatial Gaussian process can be directly derived from a spatial linear Gaussian process $Y(\mathbf{s}) = \mu + \omega(\mathbf{s})$ (see Jona Lasinio, Gelfand, and Jona Lasinio (2012) G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) and Jona Lasinio, Santoro, and Mastrantonio (2020)).
$\omega(\mathbf{s})$ is a univariate Gaussian process with mean zero and covariance matrix defined by a spatial variance parameter σ^2 combined with a correlation function $R(h_{sp}; \boldsymbol{\rho})$ parametrized with $\boldsymbol{\rho}$ and with distances between spatial locations h_{sp} as inputs.

A sample of n realizations of this process follows a multivariate Gaussian distribution as described before and is given by $\mathbf{Y} = (Y_1, Y_2, ..., Y_n) \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 R(., \boldsymbol{\rho}))$ where $\boldsymbol{\mu} = (\mu(\mathbf{s}_1), \mu(\mathbf{s}_2), ..., \mu(\mathbf{s}_n))$ with a mean function $\mu(.)$. The spatial covariance matrix $\sigma^2 R(., \boldsymbol{\rho})$ is generated with a covariance function which describes the structure of spatial dependence between different locations e.g. $\kappa(\mathbf{s}_j, \mathbf{s}_k; \boldsymbol{\rho}) = \sigma^2 R(\boldsymbol{\rho})_{jk} = \sigma^2 R(||\mathbf{s}_j - \mathbf{s}_k||; \boldsymbol{\rho})$. Then, $\Theta \sim \mathcal{WN}(\boldsymbol{\mu}, \sigma^2 R(., \boldsymbol{\rho}))$ is given by

[2] The case of $p = 2$ would be used for the projected Gaussian process.
[3] $d = 3$ would be chosen for spatio-temporal setups (cf. Jona Lasinio, Santoro, and Mastrantonio (2020))

the induced wrapped process.

Bayesian Inference & Spatial Interpolation (Kriging) In the field of geostatistics, one main motivation of spatial process modeling is spatial prediction, also called spatial interpolation or kriging (see e.g. Ch. 6 in Diggle and Ribeiro jr. (2007) and Rasmussen and Williams (2006), p.30). The general idea of kriging is using observed data at a given set of locations to interpolate data at unobserved locations making use of spatial dependence structures.

For example, one might be interested in predicting the wind direction $\theta(\mathbf{s}_0)$ at a new location $\mathbf{s}_0$ given observed wind directions $\boldsymbol{\Theta}$. Obtaining such predictions in a Bayesian setup is called Bayesian kriging and comes down to the predictive distribution of $\theta(\mathbf{s}_0)$ given the observed data stored in $\boldsymbol{\Theta}$ (see Ch. 7 in Diggle and Ribeiro jr. (2007) for details on Bayesian spatial prediction). As pointed out by G. Jona Lasinio, Gelfand, and Jona Lasinio (2019), this predictive distribution is not available in closed form but can be sampled from using Markov chain Monte Carlo (MCMC) methods.

Extending the results in Jona Lasinio, Gelfand, and Jona Lasinio (2012) for a posterior mean, Mastrantonio, Jona Lasinio, and Gelfand (2016) establish a complete predictive distribution $g(\theta(\mathbf{s}_0)|\boldsymbol{\Theta})$ for the interpolation of directions at the unobserved location $\theta(\mathbf{s}_0)$ given observed directions $\boldsymbol{\Theta}$. Following the setup in Mastrantonio, Jona Lasinio, and Gelfand (2016) and G. Jona Lasinio, Gelfand, and Jona Lasinio (2019), for a set of n observed locations $\mathcal{S} \subset \mathbb{R}^2$, a vector of observed directions $\boldsymbol{\Theta} = \{\theta(\mathbf{s}), \mathbf{s} \in \mathcal{S}\}$ again corresponds to a vector of linear variables $\mathbf{Y} = \{Y(\mathbf{s}), \mathbf{s} \in \mathcal{S}\}$ with a vector of winding numbers $\mathbf{K} = \{K(\mathbf{s}), \mathbf{s} \in \mathcal{S}\}$ via the relations formalized in $\boldsymbol{\Theta} = \mathbf{Y} \mod 2\pi$ and $\mathbf{Y} = \boldsymbol{\Theta} + 2\pi\mathbf{K}$.

As introduced in Mastrantonio, Jona Lasinio, and Gelfand (2016), Mastrantonio, Gelfand, and Jona Lasinio (2016) and G. Jona Lasinio, Gelfand, and Jona Lasinio (2019), the distribution $h(\theta(\mathbf{s}_0), K(\mathbf{s}_0))$ conditional on the observed directions $\boldsymbol{\Theta}$ can be formulated by aggregating over all $K(\mathbf{s})$ with $\mathbf{s} \in \mathcal{S}$:

$$h(\theta(\mathbf{s}_0), K(\mathbf{s}_0)|\boldsymbol{\Theta}) = \sum_{K(\mathbf{s}),\mathbf{s}\in\mathcal{D}} \int_{\boldsymbol{\Psi}_Y} h(\theta(\mathbf{s}_0), K(\mathbf{s}_0)|\boldsymbol{\Psi}_Y, \mathbf{K}, \boldsymbol{\Theta}) \times h(\boldsymbol{\Psi}_Y, \mathbf{K}|\boldsymbol{\Theta})\, d\boldsymbol{\Psi}_Y \qquad (6)$$

where $\boldsymbol{\Psi}_Y$ is a vector of parameters of the multivariate density $h(.)$ of $\mathbf{Y}$ in the same sense as e.g. $\boldsymbol{\eta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ from section 1 for $h(., \boldsymbol{\eta})$ being the multivariate Gaussian distribution (cf. Mastrantonio, Jona Lasinio, and Gelfand (2016), p.335).

To get samples for $\theta(\mathbf{s}_0)$, Mastrantonio, Jona Lasinio, and Gelfand (2016) and Mastrantonio, Gelfand, and Jona Lasinio (2016) show how to make use of equation (6) by first sampling some K^l and Ψ^l from their posterior distribution $h(\mathbf{\Psi}_Y, \mathbf{K}|\mathbf{\Theta})$ where $l = 1, 2, ..., L$ and L is the total number of MCMC samples to be taken. Subsequently, $\theta^l(\mathbf{s}_0)$ is sampled from $h(\theta(\mathbf{s}_0), K(\mathbf{s}_0)|\mathbf{\Psi}_Y^l, \mathbf{K}^l, \mathbf{\Theta})$ for each of these posterior samples K^l and Ψ^l. By this iterative procedure called composition sampling, each sample $\theta(\mathbf{s}_0)^l$ can be seen as a sample from $h(\theta(\mathbf{s}_0), K(\mathbf{s}_0)|\mathbf{\Theta})$. Keeping only the results for $\theta(\mathbf{s}_0)$, one can obtain a set $\{\theta^l(\mathbf{s}_0), l = 1, 2, ..., L\}$ of samples from the predictive distribution (cf. Mastrantonio, Jona Lasinio, and Gelfand (2016) and G. Jona Lasinio, Gelfand, and Jona Lasinio (2019)).

For sampling $\theta^l(s_0)$ from $h(\theta(s_0), K(s_0)|\mathbf{\Psi}_Y^l, \mathbf{K}^l, \mathbf{\Theta})$, Mastrantonio, Jona Lasinio, and Gelfand (2016) point again to the correspondence of the wrapped variable to a linear one, namely $\mathbf{Y} = \mathbf{\Theta} + 2\pi\mathbf{K}$. Because of this connection, sampling each $\theta^l(\mathbf{s}_0)$ from $h(\theta(s_0), K(s_0)|\mathbf{\Psi}_Y^l, \mathbf{K}^l, \mathbf{\Theta})$ is in fact equivalent to sampling $Y^l(s_0)$ from $h(Y(s_0)|Y^l, \Psi^l)$ for each $l = 1, 2, ..., L$ (cf. Mastrantonio, Jona Lasinio, and Gelfand (2016), p.335).

Then, the sampling of $Y^l(s_0)$ makes use of the fact that $\mathbf{Y} = (Y_1, Y_2, ..., Y_n)^T$ stems from a linear Gaussian process, i.e. with the definition of Gaussian processes provided in section 2.1.5, this means that a given finite subset of this linear Gaussian process follows a joint multivariate Gaussian distribution and $\mathbf{Y} \sim \mathcal{N}(\mu_Y \mathbb{1}_n, \sigma^2 R(., \boldsymbol{\rho}))$ as introduced before. Using the properties of multivariate Gaussian distributions revisited in section 2.1.2, the joint distribution of $Y(\mathbf{s}_0)$ and $\mathbf{Y}$ conditioned on $\mathbf{\Psi}_Y$ for a new location $\mathbf{s}_0$ is given by:

$$\begin{pmatrix} \mathbf{Y}(\mathbf{s}_0) \\ \mathbf{Y} \end{pmatrix} | \mathbf{\Psi}_Y \sim \mathcal{N}\left(\begin{pmatrix} \mu_Y \\ \mu_Y \mathbb{1}_n \end{pmatrix}, \sigma_Y^2 \begin{pmatrix} 1 & R(., \boldsymbol{\rho})_\star^T \\ R(., \boldsymbol{\rho})_\star & R(., \boldsymbol{\rho})_Y \end{pmatrix} \right) \tag{7}$$

where $(R(., \boldsymbol{\rho})_Y)_{ij} = R(||\mathbf{s}_i - \mathbf{s}_j||; \boldsymbol{\rho})$ and $(R(., \boldsymbol{\rho})_\star)_i = (R(., \boldsymbol{\rho})_{Y, Y(\mathbf{s}_0)})_i = R(||\mathbf{s}_i - \mathbf{s}_0||; \boldsymbol{\rho})$ in the covariance matrix of the joint distribution.

As presented in Mastrantonio, Jona Lasinio, and Gelfand (2016) and Mastrantonio, Gelfand, and Jona Lasinio (2016), it is possible to derive the distribution of $Y(\mathbf{s}_0)|\mathbf{Y}, \mathbf{\Psi}_Y$ from the above joint distribution of $Y(\mathbf{s}_0), \mathbf{Y}|\mathbf{\Psi}_Y$. Using the properties of multivariate Gaussian distributions, namely that they are closed under marginalization and conditioning, as well as results for obtaining a conditional Gaussian distribution from a joint Gaussian distribution from section 2.1.2, one gets $Y(\mathbf{s}_0)|\mathbf{Y}, \mathbf{\Psi}_Y \sim \mathcal{N}(M_{Y(\mathbf{s}_0)}, Var_{Y(\mathbf{s}_0)})$ with mean $M_{Y(\mathbf{s}_0)}$ and variance $Var_{Y(\mathbf{s}_0)}$ where

$$M_{Y(\mathbf{s}_0)} = \mu_Y + \sigma_Y^2 R(\boldsymbol{\rho})_\star^T (\sigma_Y^2 R(\boldsymbol{\rho})_Y)^{-1} (\mathbf{Y} - \mu_Y \mathbb{1}_n) \tag{8}$$

$$\text{and } Var_{Y(\mathbf{s}_0)} = \sigma_Y^2 - \sigma_Y^2 R(\boldsymbol{\rho})_\star^T \left(\sigma_Y^2 R(\boldsymbol{\rho})_Y\right)^{-1} \sigma_Y^2 R(\boldsymbol{\rho})_\star. \tag{9}$$

Consequently, one can iteratively use the l-th sample to get a mean $M^l_{Y(\mathbf{s}_0)}$ and a variance $Var^l_{Y(\mathbf{s}_0)}$ with $l = 1, 2, ..., L$ and use these results to sample each $Y^l(\mathbf{s}_0)$ from $\mathcal{N}(M^l_{Y(\mathbf{s}_0)}, Var^l_{Y(\mathbf{s}_0)})$. Then, $\theta^l(\mathbf{s}_0) = Y^l(\mathbf{s}_0) \bmod 2\pi$ is a posterior sample from the predictive distribution and as such predictive inference for the circular variable can be obtained using the linear $Y^l(\mathbf{s}_0)$ for each of the $l = 1, 2, ..., L$ MCMC samples.

These results are presented in Mastrantonio, Gelfand, and Jona Lasinio (2016) and Mastrantonio, Jona Lasinio, and Gelfand (2016) in greater detail. In the recent work by Jona Lasinio, Santoro, and Mastrantonio (2020), the authors provide an additional step-by-step explanation of the implementation of the wrapped spatial Gaussian process model in their `R`-package `CircSpaceTime` (G. Jona Lasinio, Mastrantonio, and Santoro 2019). In particular, they show how MCMC methods are used to obtain posterior samples and offer in-depth explanations of the update of the mean parameter by Gibbs sampling and the two covariance parameters that are obtained together via the implementation of the Metropolis-Hastings algorithm (see Jona Lasinio, Santoro, and Mastrantonio (2020) for more details on the implemented algorithms).

Important Extensions Even though, the focus is set on the wrapped Gaussian process for spatial prediction of circular data here, it is important to note that there are a lot of interesting extensions to wrapped Gaussian process models available as well as other approaches like projected Gaussian processes that formulate circular extensions for Gaussian processes.

For example, starting from Jona Lasinio, Gelfand, and Jona Lasinio (2012) and Wang and Gelfand (2014), wrapped and projected spatial Gaussian processes are extended to spatio-temporal circular models by Mastrantonio, Jona Lasinio, and Gelfand (2016) allowing for the simultaneous analysis of directional data connected over space and time. Using the Gneiting correlation function as introduced by Gneiting (2002), they provide results for a spatio-temporal Gaussian process models for circular variables that are also implemented in `CircSpaceTime` (G. Jona Lasinio, Mastrantonio, and Santoro 2019). G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) present an additional comparison of spatial and spatio-temporal projected and wrapped Gaussian processes illustrating how a spatio-temporal wrapped Gaussian process is induced by the corresponding spatio-temporal linear Gaussian process.

Mastrantonio, Gelfand, and Jona Lasinio (2016) introduce wrapped skewed Gaussian processes that allow for asymmetric marginal distributions in the wrapped Gaussian process for a spatial and spatio-temporal setup. They show that a wrapped skewed Gaussian process can also be obtained from a linear skewed Gaussian process by the previously used modulo transformation. Using simulated data and a real-data example with wave directions as in Jona Lasinio, Gelfand, and Jona Lasinio (2012), they point out that out-of-sample prediction with wrapped skewed Gaussian processes improves compared to wrapped Gaussian process models for spatial interpolation.

Moreover, directional measurements like wave or wind directions can be assumed to vary dependent on other linear variables like wave height or wind speed. Therefore, incorporating linear influential variables in spatial Gaussian processes models for circular data is another field of ongoing research (cf. G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) and Jona Lasinio, Santoro, and Mastrantonio (2020)).
Wang, Gelfand, and Jona Lasinio (2015) provide detailed explanations of such joint modeling approaches for a directional and a linear variable with wave directions and wave heights and present spatial interpolation as well as spatio-temporal prediction in such a joint setup. G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) provide an additional concise overview of these joint models for circular and linear data and explain how they build joint models using projected Gaussian processes for circular measurements that are conditioned on linear spatial processes for the linear variable.

For now, these extensions are left aside shifting the focus back to the wrapped Gaussian Process models previously introduced in this section.
The following section evaluates hyperprior sensitivity and predictive performance of wrapped Gaussian process models under different hyperprior settings and sample sizes for training and test data, thus further exploring wrapped Gaussian process models as implemented in `CircSpaceTime` (G. Jona Lasinio, Mastrantonio, and Santoro 2019). Moreover, a real-data application of wrapped Gaussian process models for spatial interpolation of wind directions in Germany is presented in section 4.

3 Hyperpriorsensitivity of wrapped GP models

The following part uses simulated data to explore how the accuracy of spatial interpolation is affected by different hyperprior settings as well as different sample sizes in training and test data. The simulation examples partly follow Jona Lasinio, Gelfand, and Jona Lasinio (2012) and extend the settings tested therein.

The results for simulated data in this section and real-data examples in section 4 are obtained using the wrapped Gaussian process models and evaluation tools implemented in the `R`-package `CircSpaceTime` (G. Jona Lasinio, Mastrantonio, and Santoro 2019) that enables spatial and spatio-temporal modeling of circular data in a hierarchical Bayesian modeling framework (see Jona Lasinio, Santoro, and Mastrantonio (2020) for a detailed introduction of the implementations in `CircSpaceTime`).
A wide range of other `R`-packages is available for analyzing circular data. Jona Lasinio, Santoro, and Mastrantonio (2020) offer an extensive overview of `R`-packages for different aspects of circular data analysis. In this work, apart from `CircSpaceTime`, the `R`-packages `circular` (Lund et al. 2017) and `CircStats` (Lund and Agostinelli 2018) are used for summaries and descriptive analyses of the circular measurements.

3.1 Motivation

Loosely speaking, in contrast to frequentist approaches, Bayesian modeling explicitly works with prior beliefs and their influence on estimation and prediction. This dependency can be seen as a powerful and transparent way to connect modeling efforts and existing knowledge or assumptions but can also constitute a limitation when, for example, results are entirely driven by prior assumptions. In consequence, it is often interesting to explore how sensitive posterior inference of a given modeling setup is to the prior assumptions (see Gelman et al. (2014)).

The focus of the following simulation is to investigate hyperpriorsensitivity of wrapped Gaussian process models by comparing different combinations of hyperparameters in the priors while controlling for the potential effects of different sample sizes of training and test datasets. The goal is to explore the robustness of the models and in particular whether and how accuracy measures differ for different setups.

3.2 Modeling Setup in `CircSpaceTime`

This simulation study uses the function `CircSpaceTime::WrapSp()` to sample from the posterior distribution of a spatial wrapped Gaussian model. For each MCMC chain, posterior samples for the circular mean are stored in `alpha`, `sigma2` for the variance and `rho` for the spatial correlation decay parameter.
With the posterior samples obtained and a given set of test data at unobserved locations, the function `CircSpaceTime::WrapKrigSp()` is used for posterior estimation and spatial interpolation on this given set of unobserved locations. `CircSpaceTime::WrapKrigSp()` returns posterior spatial predictions as well as the mean and variance of the associated linear Gaussian process on the unobserved locations over all posterior samples (cf. G. Jona Lasinio, Mastrantonio, and Santoro (2019) and Jona Lasinio, Santoro, and Mastrantonio (2020)).

As explained in Jona Lasinio, Gelfand, and Jona Lasinio (2012), model fitting of a wrapped spatial Gaussian process model starts with defining a corresponding linear spatial Gaussian process making use of the correspondence of a linear and a wrapped process, i.e. $\mathbf{\Theta} = \mathbf{Y} + 2\pi\mathbf{K}$. As introduced before, one can assume a spatial linear Gaussian process model $Y(\mathbf{s}_i) = \mu + \omega(\mathbf{s}_i)$ with each $\mathbf{s}_i \in \mathbb{R}^2$ and $i = 1, 2, ..., n$. In this spatial setup, $\omega(s_i)$ denotes a Gaussian process with zero mean and covariance structure described by $\sigma^2 R(||\mathbf{s}_i - \mathbf{s}_j||; \rho)$ where σ^2 denotes the process variance, ρ is the spatial decay parameter and R is a spatial correlation function.

Referring to Jona Lasinio, Gelfand, and Jona Lasinio (2012), the following simulation assumes an exponential correlation structure $R(h_{sp}; \rho) = exp(-\rho\ h_{sp})$ with spatial decay parameter ρ and spatial distances h_{sp} by choosing `corr_fun = "exponential"` in the function `CircSpaceTime::WrapSp()`.
Here, the focus is set on different hyperparameter specifications in wrapped Gaussian process models rather than testing different kernel functions. Therefore, only the exponential kernel $exp(-\rho h_{sp})$ is considered in the simulation and in the real-data examples presented below.

As in Jona Lasinio, Gelfand, and Jona Lasinio (2012), wrapped Gaussian process models in the following simulated examples are fitted with training data where for each replicate in each setting `CircSpaceTime::WrapSp()` creates 2400 posterior samples using $30,000$ MCMC iterations with a burn-in of 6000 and a thinning factor of 10.

The process variance `sigma2` and the spatial decay parameter `rho` are sampled jointly using an adaptive Metropolis step where `sd_prop=list("sigma2"=0.1,"rho"=0.1)` sets initial guesses for these variables and an acceptance ratio of 0.234 is chosen following again Jona Lasinio, Gelfand, and Jona Lasinio (2012).
Moreover, the argument `adapt_param=c(start=40000,end=45000,exp=0.5)` allows to control the start, end and speed of the adaption algorithm.

As suggested in Jona Lasinio, Gelfand, and Jona Lasinio (2012), one can choose a vector of zeros as starting values for the winding number `k` when the data is centered around π. In Jona Lasinio, Gelfand, and Jona Lasinio (2012), other starting values are set to 0.021 or 0.013 for the spatial decay parameter (denoted ϕ in Jona Lasinio, Gelfand, and Jona Lasinio (2012))[4] and to π for the circular mean. The authors also test three different starting values for the variance, namely $\{0.1, 0.5, 1\}$.
In contrast, the following simulation setup uses `alpha` $= \pi$, `sigma2` $= 0.5$ and `rho` $= 0.021$ as starting values for all models while comparing different weakly informative and informative settings for the hyperparameters of the assumed prior distributions.

3.2.1 Hyperprior Assumptions in `CircSpaceTime`

For the three parameters (μ, σ^2, ρ) of the model described above, certain prior distributions are assumed in `CircSpaceTime`. Following Jona-Lasinio et al. (2012), the mean μ and the variance σ^2 (denoted `sigma2` in `CircSpaceTime`) are assumed to be independent as well as distributed according to a Gaussian distribution and an Inverse Gamma distribution, respectively. In `CircSpaceTime`, the assumed Gaussian prior on μ is not set directly but instead the corresponding circular mean denoted `alpha` is specified with an induced wrapped Gaussian prior. Moreover, a Uniform prior is assumed for the spatial decay parameter ρ (denoted `rho` in `CircSpaceTime`).
In particular, the prior setup in the `WrapSp()` -function of `CircSpaceTime` assumes a wrapped Gaussian distribution for the mean `alpha` $\sim \mathcal{WN}(\mu_\alpha, \sigma_\alpha^2)$, an Inverse Gamma distribution for the variance `sigma2` $\sim$ Inv $\Gamma(s, r)$ and a Uniform distribution for the spatial decay parameter `rho` $\sim \mathcal{U}(min, max)$.

[4] To clarify, referring to Jona Lasinio, Santoro, and Mastrantonio (2020), $R(.;\rho)$ denotes the correlation function with spatial decay parameter ρ as introduced above while `rho` refers to this spatial decay in `CircSpaceTime`. Note that in Jona Lasinio, Gelfand, and Jona Lasinio (2012), the authors define the same correlation function as $\rho(.;\phi)$ with spatial decay parameter ϕ.

The wrapped Gaussian distribution was already introduced in more detail in Chapter 2. The Uniform distribution is $f(x; min, max) = \frac{1}{max-min}$ for $min \leq x \leq max$ with parameters for minimum and maximum. The Inverse Gamma distribution $f(x; s, r) = \frac{r^{-s}}{\Gamma(s)}(1/x)^{s+1} exp\left(-\frac{1}{rx}\right)$ is the reciprocal distribution of the Gamma distribution with parameters for shape and scale or shape and rate where rate = 1/scale. The shape parameter s controls the height of the function where higher values for s also lead to thinner tails of the distribution. The scale parameter controls the spread of the function. Jona Lasinio, Santoro, and Mastrantonio (2020) provide an overview over all prior distributions implemented in `CircSpaceTime` (see Table 2 therein).

3.2.2 Selecting Hyperprior settings

In the simulated data examples in Jona Lasinio, Gelfand, and Jona Lasinio (2012), the authors test a Gaussian prior for μ "with zero mean and large variance" (Jona Lasinio, Gelfand, and Jona Lasinio (2012), p. 13) with an induced wrapped Gaussian prior distribution. They assume an informative Inverse Gamma prior for σ^2 centered at the true value with different assumed variances $\{0.01, 0.06, 0.07\}$ and a Uniform distribution for the spatial decay parameter in $[0.001, 1)$ for initial values of σ^2 either 0.1 or 0.5 and in $[0.001, 0.5)$ when the initial σ^2 is 1 (see Jona Lasinio, Gelfand, and Jona Lasinio (2012), p. 13).

Even though Jona Lasinio, Gelfand, and Jona Lasinio (2012) point out that all prior settings in their simulation study were tested with different variance values "to assess behavior under strongly and weakly informative priors" (Jona Lasinio, Gelfand, and Jona Lasinio (2012), p. 13), they focus on results for different starting values rather than comparing these different hyperparameter settings.

In Jona Lasinio, Santoro, and Mastrantonio (2020), the authors show real-data examples using wind direction measurements and explore the influence of different variability in the given data of wind directions in Italy. Therein, the authors come to the conclusion that "the wrapped model is not very sensitive" (Jona Lasinio, Santoro, and Mastrantonio (2020), p.1328). For the wrapped Gaussian process model, four different settings for minimum and maximum of `rho`'s Uniform prior are tested in `R`-code of the paper provided by the authors where the minimum of `rho`'s Uniform prior is defined anti-proportional to maximum distances between all available locations in the wind direction data and the value chosen for the maximum is obtained anti-proportional to the minimum distances between locations. The authors report that the selected different ranges between minimum and maximum of the

Uniform prior affect the time that the wrapped Gaussian process models take to converge but that all tested settings work for the wrapped Gaussian process model.

There are six hyperparameters that will be varied in the following simulation study. Namely, different parameter values are explored for the mean μ_α and variance σ_α^2 of the wrapped Gaussian prior, for shape s and rate r of the Inverse Gamma prior and for the minimum and maximum of ρ's Uniform prior.
In general, a weakly informative setup is assumed to correspond to a mean unequal to the true value and to a large variance in the wrapped Gaussian distribution for `alpha`, to an Inverse Gamma with a small value for shape = scale for `sigma2` and to a large distance between the minimum and maximum of the Uniform prior on `rho`.
An informative hyperprior setting is using a mean equal or close to the true value and a small variance in the wrapped Gaussian distribution for `alpha` and e.g. shape = 1 and scale = 5, i.e. rate = $1/5 = 0.2$, for the Inverse Gamma as well as a small range for the Uniform prior on `rho`.

Different weakly informative and informative hyperprior settings were tested here by iterating over different lists of assumed specifications of the hyperparameters for the prior distributions for `alpha`, `sigma2` and `rho` within the argument `prior` in the function `CircSpaceTime::WrapSp()`. As a starting point, six different hyperprior settings are selected and investigated in section 3.5.1 below. Building up on the results, the setup is extended in section 3.5.2. The selected hyperparameter values for the exploratory first step and the extended setup can be found in Table 1 and 3, respectively.

3.3 Steps of Simulation

The simulated examples presented here are run using compute servers provided by the Humboldt Lab of Empirical and Quantitative Research (LEQR). The general setup of the simulation study in `R` (R Core Team 2020) is largely inspired by the work of Peikert (2019) who shows examples of working with Makefiles for running simulations in parallel as well as the parallelization approaches presented in Lee, Sriutaisuk, and Kim (2020). The `R`-package `purrr` (Henry and Wickham 2020) as well as tools from the `tidyverse` (Wickham 2019) are used to enable parallel computation.
The procedure of the simulation is as follows. To get results for different combinations of hyperprior settings while controlling for different training and test data sample sizes, a function

is built to iterate over a table of all combinations of settings created using `tidyr::crossing()` from the R-package `tidyr` (Wickham 2020). In particular, for the first exploratory step, we create a grid of all combinations of settings via:

```
if(!requireNamespace("pacman"))install.packages("pacman")
pacman::p_load(tidyverse)
# true values
truealpha <- pi
truesigma2 <- 0.5
truerho <- 0.021
# sample sizes for training data
sample_sizes <- c(50, 100, 250, 500)
# sample sizes for validation set for all settings
val_sample_sizes = c(500, 1000)
# replicates
nr_iterations <- 1:100
# placeholder for hyperprior settings
hypersettings <- c("Setting 1: weakly informative",
                   "Setting 2: weakly informative",
                   "Setting 3: weakly informative",
                   "Setting 4: informative",
                   "Setting 5: informative",
                   "Setting 6: informative")
# creating a grid of simulation conditions
conditions <- crossing(
  truealpha = truealpha,
  truesigma2 = truesigma2,
  truerho = truerho,
  hyperprior = hypersettings,
  N = sample_sizes,
  val_N = val_sample_sizes,
  iteration = nr_iterations)
```

3.3.1 Training and Test Data

In Jona Lasinio, Gelfand, and Jona Lasinio (2012), the authors generated samples of size $n = 100$ that were then split up into samples of size $n = 30$ and $n = 70$ using the leftover data as test data to test two different sizes of training and test data. In contrast, in this simulation a larger variety of different training and test data sample sizes is compared while also creating 100 replicates for each of the combinations of hyperprior settings, training and test data sample sizes. Training data sample sizes for all simulated examples are `N` $= \{50, 100, 250, 500\}$. Test data for spatial interpolation is created with sample sizes of `val_N` $= \{500, 1000\}$ (and `val_N` $= \{50, 500, 1000\}$ in the extended setup) from observations that are not used in training.
This results in a total number of 100×all combinations of `N, val_N, hyperprior` models for a simulation run, e.g. there are $100 \times 4 \times 2 \times 6 = 4800$ models in the exploratory setup in section 3.3.

To ensure a more realistic setup for spatial interpolation, possible locations are restricted to a grid of points created in Universal Transverse Mercator (UTM) coordinates covering a part of the UTM zone 32 of the Northern hemisphere. Then, locations for training and testing are sampled from this common area for each replicate in each setting while ensuring that locations that are used for training are not used for testing and spatial interpolation.
This is facilitated with a custom function that converts given coordinates from longitude and latitude to a UTM format. The complete steps of data generation can be found in `sim_gendata_grid.R`.

Then, samples from a linear (unwrapped) Gaussian process are simulated by sampling from a multivariate Gaussian distribution following Coles (1998), Coles and Casson (1998) and Jona Lasinio, Gelfand, and Jona Lasinio (2012) using the custom function `rmnorm()` introduced in the vignette of G. Jona Lasinio, Mastrantonio, and Santoro (2019) which allows for Cholesky decomposition.
Following the steps in examples in the vignette of G. Jona Lasinio, Mastrantonio, and Santoro (2019), training and test coordinates are used in the creation of a distance matrix that contributes to the covariance matrix of the unwrapped spatial Gaussian process.
A circular Gaussian process is created by wrapping these samples componentwise using $\Theta_i = Y_i \bmod 2\pi$ for the i-th entry. This is repeated until the selected sample size of `N` + `val_N` is reached where `N` and `val_N` denote the selected training and test sample sizes, respectively.

The sampled circular data is then split into a training and test dataset according to the selected `N` and `val_N`.

A reduced version of these steps is presented below, see `sim_gendata_grid.R` for the complete procedure in the simulation runs.

```
# requires rmnorm(), coordinates, true values to run
#----dist-matrix----
joint_coords <- rbind(coords, valcoords)
# create dist matrix for all coords
Dist <- as.matrix(dist(joint_coords))
#----cov-function----
# exponential covariance function with parameters sigma2 & rho
SIGMA <- sigma2 * exp(-rho*Dist)
#----simulate-data-unwrapped-GP----
Y <- rmnorm(1, rep(alpha, times=NROW(joint_coords)), SIGMA)
#----wrap-data----
theta <- c()
for(j in 1:NROW(joint_coords)) {
  theta[j] <- Y[j]%%(2*pi)
}
# create training and validation set for the directional data input
rows_theta_train <- sample(seq_len(NROW(joint_coords)), N)
thetadata <- theta[rows_theta_train]
valthetadata <- theta[-rows_theta_train]
```

Sampling different data for training and testing for each replicate in each combination of settings would make comparisons between different hyperprior settings unclear.

To ensure comparability between the different combinations and to control for the potential influence of different training and test locations in spatial interpolation, seeds for data generation are created which are the same for each replicate in each setting. The below example from `sim_settings_grid.R` presents the general idea:

```
# requires conditions-df to run
if(!requireNamespace("pacman"))install.packages("pacman")
pacman::p_load(tidyverse)
```

```r
# set seed for creating seeds
set.seed(987654321)
# nest & unnest conditions
# to create same train- and validationset
# over iterations and sample sizes
conditions <- conditions %>%
  nest(hyperprior, N, val_N) %>%
  dplyr::mutate(
    datagenseed = as.integer(runif(length(iteration)) * 1e6)
    ) %>%
  unnest(., data)
```

Hence, training and test data vary among the 100 replicates for each hyperprior setting. At the same time, the above procedure ensures a level of comparability between the replicates of each combination of hyperprior settings, training and test data sample sizes because training and test data in each of the 100 replicates only differ in sample sizes among the different hyperprior settings.

3.4 Tools for Model Evaluation and Comparison

Some ways to evaluate the resulting posterior samples as well as spatial interpolation accuracy are presented here for the wrapped Gaussian process models in `CircSpaceTime`.

3.4.1 Mean Point Estimates, Credible Intervals and Coverage Rates

As pointed out earlier, `CircSpaceTime::WrapSp()` gives back posterior samples for `alpha`, `sigma2` and `rho`. Using the correspondence between the linear and wrapped Gaussian process introduced above, posterior samples of `alpha` combined with the winding number `k` offer information about posterior features of μ. Moreover, the posterior information about `sigma2` gives posterior information about the concentration parameter c via $c = e^{-\sigma^2/2}$ as described earlier. To evaluate results, mean point estimates for μ and c are computed for each of the replicates in each combination of settings. Following Jona Lasinio, Gelfand, and Jona Lasinio (2012), results for the spatial decay parameter `rho` are summarized as modal values.

Additionally, one can compute credible intervals for all posterior estimates. In Bayesian inference a 95% credible interval is defined as an interval $[t_l, t_u]$ for which it holds that

$\int_{t_l}^{t_u} f(\eta|x)d\eta = 0.95$ where $f(\eta|x)$ denotes a posterior distribution of model parameter η (see e.g. Held and Sabanés Bové (2014)).
There exists an analogeous approach for circular data, namely the posterior credible arc as described in Jona Lasinio, Gelfand, and Jona Lasinio (2012), p. 7. As mentioned therein and following Fisher (1993), a symmetric posterior 95% credible arc is the arc that contains 95% of the posterior samples of a circular parameter of interest. Here, 95% credible arcs for the circular `alpha` and 95% credible intervals for non-circular `sigma2` and `rho` are calculated using the 2.5%- and the 97.5%-quantile of the given vector of posterior samples.

To assess and compare results for each combination of hyperprior settings and training and test sample sizes, mean point estimates and modal values are averaged over all 100 replicates in each setting. To summarize credible arcs and intervals in the same spirit, one can make use of coverage rates, i.e. compute a percentage of all cases in which the true value of the given model parameter is included in the credible arc or interval derived from the given posterior sample.

3.4.2 Circular Measures of Prediction Accuracy

Following Jona Lasinio, Gelfand, and Jona Lasinio (2012), predictive accuracy of the models is evaluated using circular versions of the average prediction error (APE) and the continuous ranked probability score (CRPS).
For circular data, the average prediction error is a measure of circular distance between observed and interpolated estimates using $d(a, b) = 1 - cos(a - b)$ as a measure of circular distance for two angular variables a and b (see Jammalamadaka and SenGupta (2001), Mardia and Jupp (2000) and Jona Lasinio, Gelfand, and Jona Lasinio (2012)). Following Mardia and Jupp (2000), p.18, one can measure the dispersion of angles $a_1, ..., a_n$ around a given angle b via $D(a) = \frac{1}{n}\sum_{i=1}^{n}\{1 - cos(a_i - b)\}$.
This formula is implemented in `CircSpaceTime::APEcirc()` that computes pointwise and mean average prediction error assessing the difference between the true values and the interpolated directions at the testset locations obtained with `CircSpaceTime::WrapKrigSp()`.

A second tool for evaluating spatial predictions of the wrapped Gaussian process models is the circular continuous ranked probability score (CRPS) introduced in Grimit et al. (2006). It provides a score comparing interpolated directions captured in a cumulative forecast distribution and true values at the testset locations.

Using the circular distance as introduced above for the circular APE, a circular CRPS can be obtained as $CRPS_{circ}(P, \theta) = \mathbb{E}[d(\Theta, \theta)] - \frac{1}{2}\mathbb{E}[d(\Theta, \Theta^\star)]$ where P is a forecast distribution, θ is a holdout value and Θ and $\Theta^\star$ are independent circular variables from the distribution P (see Grimit et al. (2006) and Jona Lasinio, Gelfand, and Jona Lasinio (2012)). The function `CircSpaceTime::CRPScirc()` computes pointwise and mean CRPS for the posterior predicted spatial directions.

3.5 Results

The following subsection presents results of the simulation examples and discusses potential implications.

3.5.1 Exploring different Hyperprior Settings

To test how wrapped Gaussian process models perform under different hyperprior settings and different sample sizes of training and test data, the first exploratory step taken here is to test six hyperprior settings, four training sample sizes $\texttt{N} = \{50, 100, 250, 500\}$ and two test sample sizes $\texttt{val\_N} = \{500, 1000\}$. Exploring these six settings enables to identify patterns of interest that are then investigated further in an extended simulation setup in section 3.5.2. The six hyperprior settings are presented in detail in Table 1. Settings 1 to 3 resemble weakly

	Setting 1	Setting 2	Setting 3	Setting 4	Setting 5	Setting 6
$\texttt{alpha} \sim \mathcal{WN}$	$(0, 2\pi)$	$(\pi/2, 2\pi)$	$(0, 2\pi)$	$(\pi, \pi/2)$	$(\pi, \pi/4)$	$(\pi, \pi/8)$
$\texttt{sigma2} \sim \text{Inv}\Gamma$	$(0.05, 0.05)$	$(0.1, 0.1)$	$(0.1, 0.1)$	$(1, 0.2)$	$(1, 0.5)$	$(1, 0.5)$
$\texttt{rho} \sim \mathcal{U}$	$(0, 10)$	$(0, 5)$	$(0, 2)$	$(0, 0.5)$	$(0.01, 0.03)$	$(0.018, 0.028)$

Table 1: Overview of hyperprior settings tested in a first exploratory simulation setup.

informative settings, while Settings 4 to 6 are examples of informative settings with respect to the true values of $\texttt{alpha} = \pi$, $\texttt{sigma2} = 0.5$ and $\texttt{rho} = 0.021$.

Figure 6 and Figure 7 show boxplots of the mean APE and mean CRPS over 100 replicates in each of the 48 combinations of hyperprior settings and sample sizes. Lower values of APE or CRPS indicate higher accuracy.

The values of mean APE in Figure 6 are less dispersed, the larger the training sample size `N`. Moreover, mean APE values in setups with $\texttt{N} = 50$ are rather similar for all the 6 different hyperprior settings.

As expected for informative settings, Setting 4 performs slightly better than the other weakly

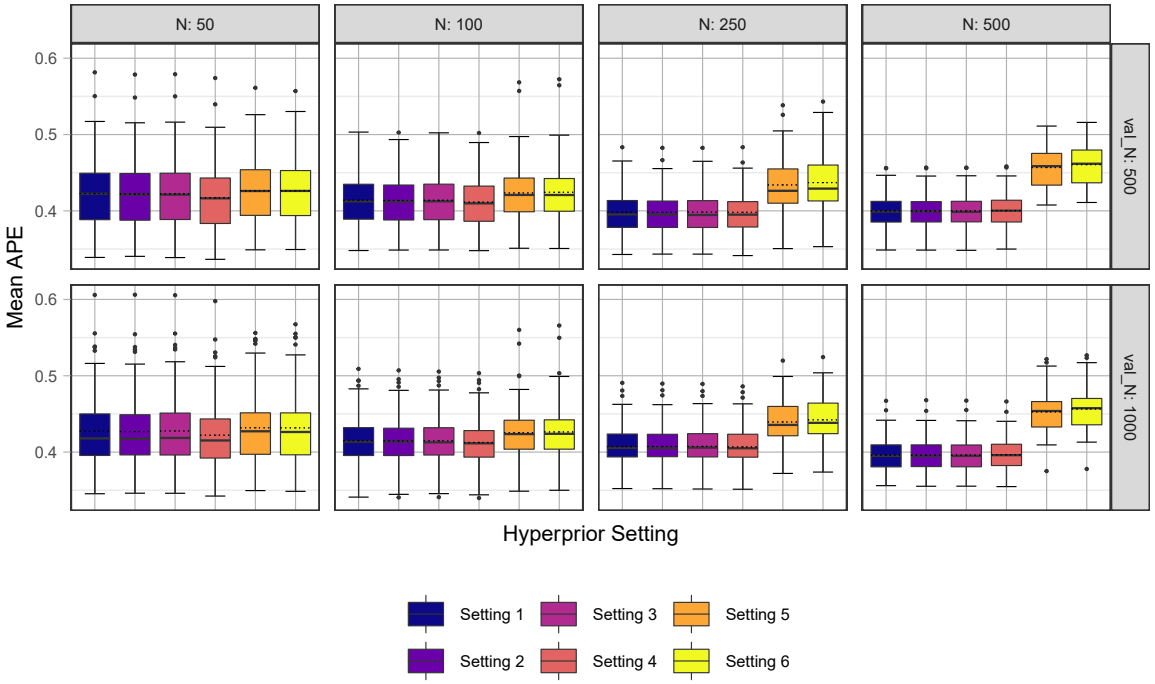


Figure 6: Boxplots of mean APE in exploratory simulation setup. Each boxplot summarizes mean APE for 100 replicates in each of the combinations of hyperprior settings and training and test sample sizes. Median values as solid lines, mean values as dashed lines.

informative settings for smaller training sample sizes $\texttt{N} = 50$ and $\texttt{N} = 100$, i.e. when less information is available through the training data, while this effect is no longer present for increasing training sample sizes. For $\texttt{N} = 250$ and $\texttt{N} = 500$, Settings 1 to 3 and Setting 4 show similar results.

Settings 5 and 6 perform slightly worse than Setting 4 when $\texttt{N} = 50$. Also, for $\texttt{N} = 50$, there is no clear difference in mean APE between Settings 5 and 6 and the weakly informative settings.

Notably, starting with $\texttt{N} = 100$ and continuing for $\texttt{N} = 250$ and $\texttt{N} = 500$, one can observe an emerging pattern where the informative Settings 5 and 6 perform increasingly worse for larger training sample sizes compared to the three weakly informative settings in Setting 1 to 3 as well as compared to the informative Setting 4.

Like mean APE values, mean CRPS over 100 replicates in each of the combinations depicted in Figure 7 are dispersed for smaller training sample sizes and more concentrated for larger training sample sizes over the 100 replicates in each combination. The general pattern over the different combinations of hyperprior settings and sample sizes is very similar between mean APE and mean CRPS.

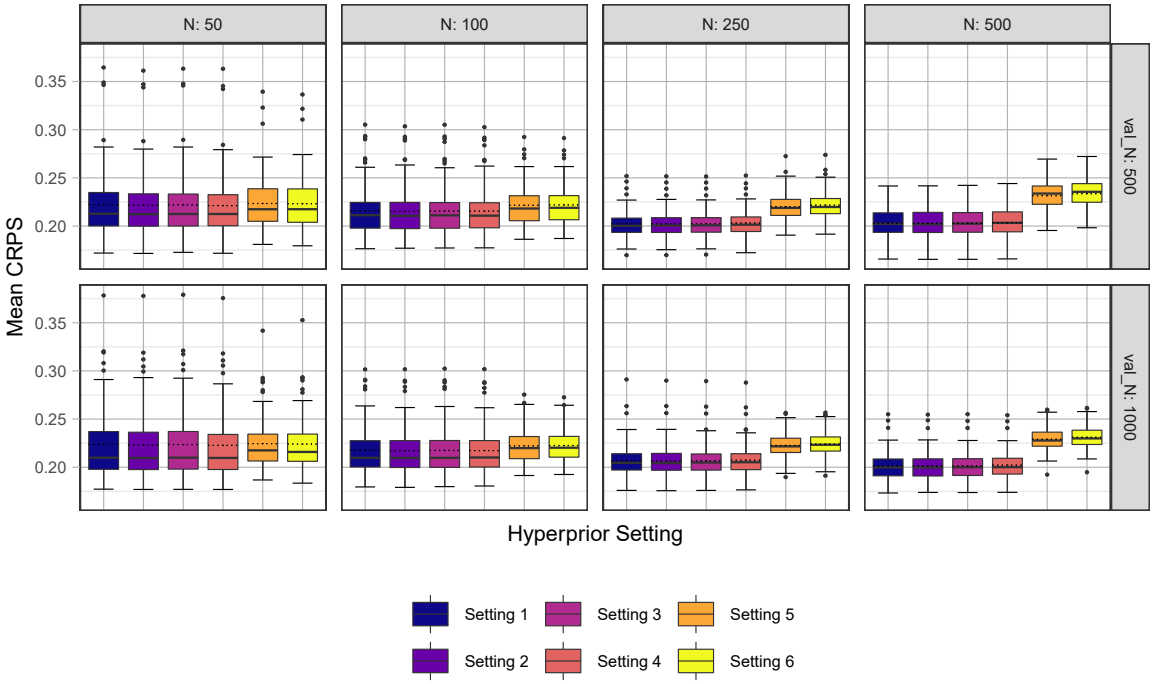


Figure 7: Boxplots of mean CRPS in exploratory simulation setup. Each boxplot summarizes mean CRPS for 100 replicates in each of the combinations of hyperprior settings and training and test sample sizes. Median values as solid lines, mean values as dashed lines.

Most notably, for mean CRPS, one can again observe that while Settings 1 to 4 show a slight decrease in mean CRPS for larger training sample sizes, mean CRPS values for Settings 5 and 6 increase.

These findings call for further clarification. The question is which properties separate Settings 5 and 6 from the other weakly informative settings as well as from the informative Setting 4 and thus might explain lower prediction accuracy in these settings for larger training sample sizes.

Referring to Table 1, one can see that only Settings 5 and 6 assume parameters $(1, 0.5)$ for the Inverse Gamma prior of `sigma2`. Moreover, Settings 5 and 6 are the only settings where `rho`'s Uniform prior is assumed to have minimum and maximum values that span a small range around the true value with minimum values $\neq 0$. All other weakly informative or informative settings assume a broader range for the Uniform distribution with $min = 0$.

As a point of reference for the results of these exploratory steps, Table 2 summarizes the mean APE of a baseline model where the mean direction in the given training set is used to predict directions at unobserved locations. Results for mean APE are averaged over all

replicates and different hyperprior settings since seeds for data generation are the same across replicates and across the hyperprior settings as explained earlier. Comparing Table 2 with

N	val_N	mean APE
50	500	0.2216970
50	1000	0.2248642
100	500	0.2227649
100	1000	0.2231387
250	500	0.2203383
250	1000	0.2220524
500	500	0.2215847
500	1000	0.2197035

Table 2: Overall mean APE of baseline model in the exploratory simulation setup with 8 different combinations of training and test sample sizes, denoted `N` and `val_N` respectively. Mean direction in training data is used as predicted direction for the given test locations.

the results presented in Figure 6, the wrapped Gaussian process models in the simulated examples perform considerably worse than this naive baseline. Keeping these comparisons in mind, the observed patterns of APE and CRPS values are now further explored.

3.5.2 Extended Simulation Setup

To test the influence of hyperprior settings especially for `sigma2` and `rho` on mean APE and CRPS values in more detail and to explore the patterns summarized in Figure 6 and Figure 7, the setup from before is extended. In particular, Table 3 shows the additional hyperprior settings that are included to gain more insights. To explore the effect of a small range in the Uniform prior of `rho`, Setting 1.1 applies an informative hyperprior setting for `rho` in an otherwise weakly informative setup for `alpha` and `sigma2` taken from Setting 1.

Moreover, Settings 5 and 6 are extended to Setting 5.1 and Setting 6.1 where the hyperprior settings for `sigma2` are taken over from the values in Setting 4 to see whether the effects on APE and CRPS values for larger training sample sizes are modified.

Also, Settings 5.2 and 6.2 aim at testing whether results change when a hyperprior setting for `rho` is assumed that defines a small range but is symmetric around the true value.

Settings 5.3 and 6.3 as well as Setting 1.2 follow up on the question whether values for mean APE and CRPS show different patterns over increasing training sample sizes if $min = 0$. Additionally, a test sample size of `val_N` $= 50$ is included to explore potential influences of smaller test sample sizes.

	`alpha` $\sim \mathcal{WN}$	`sigma2` $\sim$ InvΓ	`rho` $\sim \mathcal{U}$
Setting 1	$(0, 2\pi)$	$(0.05, 0.05)$	$(0, 10)$
Setting 1.1	$(0, 2\pi)$	$(0.05, 0.05)$	$(0.018, 0.024)$
Setting 1.2	$(0, 2\pi)$	$(0.05, 0.05)$	$(0, 0.024)$
Setting 2	$(\pi/2, 2\pi)$	$(0.1, 0.1)$	$(0, 5)$
Setting 3	$(0, 2\pi)$	$(0.1, 0.1)$	$(0, 2)$
Setting 4	$(\pi, \pi/2)$	$(1, 0.2)$	$(0, 0.5)$
Setting 5	$(\pi, \pi/4)$	$(1, 0.5)$	$(0.01, 0.03)$
Setting 5.1	$(\pi, \pi/4)$	$(1, 0.2)$	$(0.01, 0.03)$
Setting 5.2	$(\pi, \pi/4)$	$(1, 0.5)$	$(0.01, 0.032)$
Setting 5.3	$(\pi, \pi/4)$	$(1, 0.5)$	$(0, 0.032)$
Setting 6	$(\pi, \pi/8)$	$(1, 0.5)$	$(0.018, 0.028)$
Setting 6.1	$(\pi, \pi/8)$	$(1, 0.2)$	$(0.018, 0.028)$
Setting 6.2	$(\pi, \pi/8)$	$(1, 0.5)$	$(0.018, 0.024)$
Setting 6.3	$(\pi, \pi/8)$	$(1, 0.5)$	$(0, 0.024)$

Table 3: Hyperprior settings tested in extended simulation setup.

Results: APE and CRPS In order to investigate the prediction accuracy measures in the extended simulation setup, Figure 8 visualizes the results for mean APE for each of the 100 replicates of each of the 168 different combinations of hyperprior settings and training and test sample sizes. Looking at mean APE values summarized in Figure 8, the spreading of accuracy values again differs between different sample sizes of the training set. As before, fewer observations in the training set with `N` $= 50$ correspond to a larger dispersion in the accuracy measures, whereas APE values are more concentrated around the mean and median for larger training samples.

Moreover, results for a small test sample size of `val_N` $= 50$ are overall more dispersed whereas results for `val_N` $= 500$ and `val_N` $= 1000$ show a rather similar level of dispersion over the different combinations.

Starting with `N` $= 100$ and becoming more perceptible for `N` $= 250$ and `N` $= 500$, some settings again perform considerably worse while others tend to slightly lower mean APE values with increasing training sample sizes.

Comparing Settings 5 and 6 and the extensions to these settings, mean APE slightly increases for choosing a small range symmetric around the true value for the Uniform prior for `rho` as in Setting 6.2 as well as for a hyperprior setting with $min = 0$ as in Setting 6.3.

There is no visually detectable effect when using the values in Setting 4 for defining the

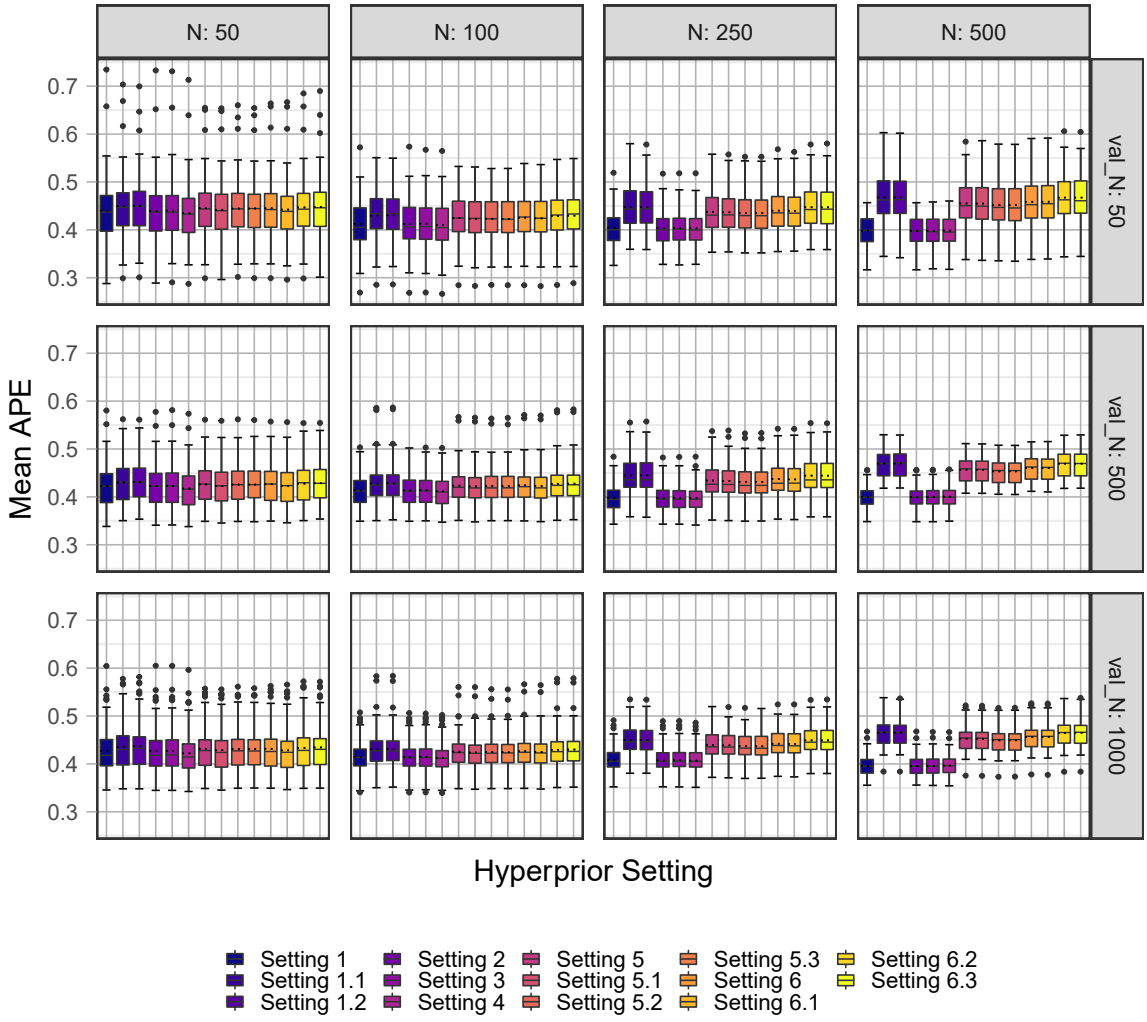


Figure 8: Boxplots of mean APE for the extended simulation setup. Each boxplot summarizes mean APE for the 100 replicates in each of the combinations of hyperprior settings and training and test sample sizes. Median values as solid lines, mean values as dashed lines.

hyperprior settings for `sigma2` instead of the values selected in Settings 5 and 6 (see Settings 5.1 and 6.1).

Interestingly, Setting 1.1 and Setting 1.2 show a very similar pattern compared to Settings 5 and Setting 6 and their respective extensions, i.e. mean APE values increase for larger training sample sizes.

As observed earlier for the exploratory step, in this extended simulation setup mean APE and mean CRPS follow the same patterns, i.e. mean CRPS increases for larger training sample sizes and Setting 1.1, Setting 1.2 and Settings 5 and 6 and their respective extensions. Similar boxplots for mean CRPS values can be found in Figure 16 in the Appendix. Another way to summarize these findings further is presented in Figure 17 and Figure 18 in the Appendix,

where the observed patterns are presented for averaged mean APE and CRPS over replicates in each setting.

Table 4 shows results for a baseline model where, again, the mean direction in the given training sample is used as a prediction for the test locations. As summarized for the ex-

N	val_N	mean APE
50	50	0.2316472
50	500	0.2216970
50	1000	0.2248642
100	50	0.2195083
100	500	0.2227649
100	1000	0.2231387
250	50	0.2212560
250	500	0.2203383
250	1000	0.2220524
500	50	0.2231624
500	500	0.2215847
500	1000	0.2197035

Table 4: Overall mean APE of baseline model in the extended simulation setup with 12 different combinations of training and test sample sizes, denoted `N` and `val_N` respectively. Mean direction in training data is used as predicted direction for the given test locations.

ploratory setup in Table 2, the mean APE valus of the baseline model are again averaged over replicates and hyperprior settings.
Mean APE values are similar to the exploratory step as the same procedure is used to generate training and test data. In comparison to mean APE values depicted in Figure 8, results of baseline models that assume the mean direction at all unobserved locations once more show lower APE values than spatial predictions obtained by the simulated examples in the extended setup.

All in all, boxplots of mean APE and mean CRPS values in Figure 8 and Figure 16 support and augment the observations from the exploratory simulation setup, i.e. for models with an assumed informative setting of the hyperparameters (and especially small ranges between *min* and *max* of `rho`'s Uniform prior) as in Settings 1.1, 1.2, 5 and 6 and their extensions, mean APE and mean CRPS values increase for increasing training sample size. This means that spatial prediction is less accurate in these cases compared to models with other informative settings as in Setting 4 or a weakly informative setting like Settings 1, 2 and 3.

Results: Mean Point Estimates In order to investigate the patterns in Figure 8 and Figure 16 further, Table 5 shows mean point estimates $\hat{\mu}$ and $\hat{c}$ and modal values $\hat{\rho}$ for the model parameters μ, c and ρ where μ is obtained by combining the circular mean in `alpha` and winding numbers `k` and $c = e^{-\sigma^2/2}$ as introduced in Jona Lasinio, Gelfand, and Jona Lasinio (2012). Mean and modal point estimates are averaged over the 100 replicates for each of the 168 combinations. Table 5 also includes mean APE and mean CRPS values averaged over the 100 replicates for a test sample size of `val_N` $= 500$.[5]

Table 5: Results from simulated data in the extended setup. Posterior mean estimates (m) obtained by averaging over MCMC samples and 100 replicates for each setting, decay parameter as modal value over MCMC samples and replicates. Coverage rates (covrate) give percentage of true values contained in 95%-credible intervals of the given posterior. True values: alpha $= \pi$, concentration $= 0.7788$ and spatial decay $= 0.021$.

	mean $\hat{\mu}$		concentration $\hat{c}$		spatial decay $\hat{rho}$		averaged for `val_N` $= 500$	
N	m	covrate	m	covrate	m	covrate	APE	CRPS
Setting 1								
50	3.142831	0.93	0.776969	0.99	5.036792	0.02	0.422736	0.222375
100	3.139120	0.89	0.778203	0.95	5.067421	0.01	0.413827	0.215604
250	3.138455	0.86	0.780299	0.90	5.084018	0.00	0.398273	0.202433
500	3.142555	0.71	0.778406	0.90	5.504697	0.00	0.400040	0.202885
Setting 1.1								
50	3.141992	0.95	0.756542	0.88	0.022001	1.00	0.429888	0.224828
100	3.147828	0.92	0.745509	0.80	0.022143	1.00	0.427954	0.223982
250	3.142063	0.97	0.694430	0.12	0.023303	0.15	0.444852	0.225598
500	3.154850	0.97	0.631831	0.00	0.023726	0.00	0.469416	0.238369
Setting 1.2								
50	3.141727	0.97	0.750498	0.88	0.012676	0.99	0.430695	0.225628
100	3.147652	0.92	0.743536	0.79	0.022512	1.00	0.428744	0.224487
250	3.142304	0.97	0.694358	0.15	0.023161	0.13	0.444939	0.225671
500	3.154046	0.97	0.631749	0.00	0.023635	0.00	0.469565	0.238414
Setting 2								
50	3.140134	0.93	0.776509	1.00	2.613000	0.05	0.422291	0.221524
100	3.137859	0.88	0.777905	0.94	2.138538	0.01	0.413771	0.215329
250	3.138063	0.86	0.780272	0.90	2.917355	0.00	0.398193	0.202367
500	3.141932	0.71	0.778436	0.90	2.823346	0.00	0.399917	0.202835
Setting 3								
50	3.142790	0.93	0.776506	0.99	0.773189	0.08	0.422803	0.222123
100	3.139392	0.88	0.777983	0.94	0.941214	0.02	0.413889	0.215572
250	3.138371	0.88	0.780217	0.88	1.355988	0.00	0.398271	0.202437
500	3.142099	0.71	0.778268	0.90	0.719808	0.00	0.400077	0.202945
Setting 4								
50	3.143174	0.93	0.781759	0.98	0.293824	0.19	0.417333	0.221299
100	3.139794	0.89	0.779993	0.92	0.242297	0.06	0.411624	0.215394
250	3.139140	0.87	0.780281	0.89	0.379727	0.00	0.398158	0.203105

[5] A similar overview for the exploratory step is included in the Appendix in Table 11

Table 5: Results from simulated data in the extended setup. Posterior mean estimates (m) obtained by averaging over MCMC samples and 100 replicates for each setting, decay parameter as modal value over MCMC samples and replicates. Coverage rates (covrate) give percentage of true values contained in 95%-credible intervals of the given posterior. True values: alpha = π, concentration = 0.7788 and spatial decay = 0.021. *(continued)*

	mean $\hat{\mu}$		concentration $\hat{c}$		spatial decay $\hat{rho}$		averaged for `val_N` = 500	
N	m	covrate	m	covrate	m	covrate	APE	CRPS
500	3.142249	0.72	0.777512	0.93	0.397509	0.00	0.400422	0.203599
Setting 5								
50	3.142701	0.98	0.760486	0.93	0.025742	0.97	0.426145	0.223366
100	3.147743	0.90	0.754429	0.87	0.027063	0.81	0.423485	0.221527
250	3.140323	0.96	0.718960	0.31	0.029008	0.00	0.434106	0.220058
500	3.152514	0.91	0.670581	0.00	0.029609	0.00	0.456781	0.231517
Setting 5.1								
50	3.142185	0.95	0.765035	0.96	0.017977	0.99	0.423119	0.223614
100	3.147912	0.90	0.756757	0.87	0.027821	0.81	0.421870	0.221568
250	3.140548	0.96	0.719914	0.34	0.028577	0.00	0.433491	0.220051
500	3.152719	0.92	0.670984	0.00	0.029558	0.00	0.456525	0.231484
Setting 5.2								
50	3.142321	0.97	0.761796	0.94	0.028703	0.95	0.425685	0.223101
100	3.147702	0.91	0.756803	0.89	0.027903	0.73	0.422483	0.220977
250	3.140888	0.96	0.724648	0.39	0.031207	0.00	0.431401	0.218692
500	3.151997	0.90	0.679899	0.00	0.031639	0.00	0.453229	0.229557
Setting 5.3								
50	3.142401	0.96	0.760642	0.94	0.020986	0.96	0.425708	0.223438
100	3.147972	0.90	0.756622	0.89	0.024713	0.71	0.422488	0.220938
250	3.140524	0.96	0.724599	0.42	0.031270	0.00	0.431453	0.218704
500	3.152104	0.89	0.679920	0.00	0.031435	0.00	0.453283	0.229604
Setting 6								
50	3.142716	0.98	0.761436	0.92	0.024003	1.00	0.426380	0.223199
100	3.148216	0.91	0.752683	0.85	0.024014	0.89	0.424204	0.221944
250	3.140896	0.96	0.712460	0.25	0.027284	0.00	0.437046	0.221679
500	3.152974	0.95	0.659743	0.00	0.027650	0.00	0.460576	0.233569
Setting 6.1								
50	3.142219	0.95	0.765935	0.96	0.024708	1.00	0.423216	0.223343
100	3.148009	0.90	0.755105	0.87	0.025478	0.86	0.422591	0.221987
250	3.141062	0.96	0.713205	0.24	0.027091	0.01	0.436487	0.221619
500	3.153776	0.94	0.660115	0.00	0.027581	0.00	0.460353	0.233587
Setting 6.2								
50	3.142637	0.97	0.758051	0.91	0.020799	1.00	0.427673	0.223670
100	3.148300	0.92	0.746341	0.80	0.022754	1.00	0.426950	0.223557
250	3.142177	0.97	0.695141	0.13	0.023196	0.15	0.444327	0.225507
500	3.154759	0.97	0.632286	0.00	0.023632	0.00	0.469221	0.238386
Setting 6.3								
50	3.142221	0.98	0.752556	0.88	0.021886	1.00	0.428445	0.224566
100	3.148435	0.93	0.744478	0.79	0.021341	1.00	0.427723	0.224118
250	3.142141	0.97	0.695119	0.14	0.023459	0.14	0.444302	0.225501
500	3.154579	0.97	0.632345	0.00	0.023796	0.00	0.469266	0.238448

Over all 168 combinations, mean point estimates $\hat{\mu}$ are rather close to `alpha` $= \pi$. Also, for $\hat{c}$ and the weakly informative Settings 1, 2 and 3 as well as the informative Setting 4, mean point estimates do not differ a lot from the starting value of `sigma2` $= 0.5$ which implies $c = e^{-\sigma^2/2} = e^{-0.5/2} \approx 0.7788$. On the other hand, for `N` $= 500$ and Settings 5 and 6 as well as the corresponding extensions of these settings, one can observe mean point estimates $\hat{c}$ that considerably differ from the true value.

For the spatial decay parameter, modal point estimates $\hat{\rho}$ vary greatly between informative and weakly informative settings. One can observe considerably higher values for $\hat{\rho}$ in Setting 4, while for informative settings with an assumed small range around the true value in the Uniform prior, i.e. Settings 1.1, 1.2, 5 and 6 and their extensions, values for $\hat{\rho}$ are mostly close to the true value of `rho` $= 0.021$. Moreover, for the weakly informative Settings 1, 2 and 3, modal point estimates $\hat{\rho}$ strongly differ from the selected starting value and reflect the assumed wider range of the Uniform prior in these setups.
Table 3 shows that in these cases the selected values for minimum and maximum define a broader interval for `rho`'s assumed Uniform prior distribution.

As introduced before, coverage rates allow to average 95%-credible intervals of the posterior samples over 100 replicates in each of the 168 combinations by measuring how often the true value of a given parameter is contained in the respective 95%-credible interval of the posterior sample of that parameter in each of the 100 models per each of the 168 combinations.
Starting with $\hat{\mu}$, one can observe high overall coverage rates for all different hyperprior settings. Notably, highest coverage rates for $\hat{\mu}$ are observed for the informative Settings 1.1., 1.2, 5 and 6 and all extensions to Settings 5 and 6 over all different training sample sizes, e.g. 0.98 for Settings 5, 6 and 6.3 and training sample size `N` $= 50$ or 0.97 for Settings 1.1., 1.2, 6.2 and 6.3 with `N` $= 250$ and `N` $= 500$.
For most of the weakly informative settings and for Setting 4, coverage rates for $\hat{\mu}$ are slightly lower, especially for large training sample sizes of `N` $= 500$. The lowest coverage rates for $\hat{\mu}$ occur for Settings 1, 2 and 3 for a training sample size of `N` $= 500$.
Interestingly, for Settings 1.1. and 1.2., where a weakly informative setting for `alpha` (where `alpha` is the wrapped mean associated with μ) and `sigma2` is combined with an informative setting for `rho` (i.e. a Uniform prior with *min* and *max* defining a small range around the true value), higher coverage rates for $\hat{\mu}$ are observed also for higher training sample sizes.

For `sigma2` and `rho` which are updated jointly, coverage rates for concentration $\hat{c}$ and spatial

decay parameter $\hat{\rho}$ for Settings 1.1, 1.2, 5 and 6 and their respective extensions tend to follow similar patterns, i.e. high coverage rates for $\hat{c}$ correspond to high coverage rates for $\hat{\rho}$ and vice versa.
In contrast, for Settings 1 to 3 and Setting 4, one can see that coverage rates for $\hat{c}$ are rather high ranging around 0.93 over all different training sample sizes while coverage rates for $\hat{\rho}$ are mostly close or equal to zero.

In order to investigate potential reasons for an increase in mean APE and CRPS values for Settings 1.1., 1.2, 5 and 6 and the extended versions compared to Settings 1 to 3 and 4 as depicted in Figure 8 and Figure 16, these settings are of special interest when combined with large training sample sizes in Table 5. In contrast to the other settings, one can observe high coverage rates for $\hat{c}$ and $\hat{\rho}$ when training sample sizes are $\texttt{N} = 50$ and $\texttt{N} = 100$ and substantially lower coverage rates close or equal to zero for larger training sample sizes.
In particular, when combined with a training sample size $\texttt{N} = 250$, coverage rates for $\hat{\rho}$ presented in Table 5 are equal to zero for all settings with the only exception of Settings 1.1. and 1.2. which show non-zero but small coverage rates for $\hat{\rho}$. For the weakly informative settings and Setting 4, coverage rates for $\hat{\rho}$ are small but non-zero for $\texttt{N} = 50$ and $\texttt{N} = 100$ and equal zero when training sample sizes are set to $\texttt{N} = 250$ and $\texttt{N} = 500$.

Overall, the averaged APE and CRPS measures for test data with $\texttt{val\_N} = 500$ reported in Table 5 do not vary greatly up until the second or third decimal places between most of the 168 combinations.
Still, one can see different patterns of decreasing or increasing averaged APE and CRPS values for different hyperprior settings and training sample sizes as illustrated earlier in more detail in Figure 8 and Figure 16.

As pointed out before, especially the coverage rates for the spatial decay parameter vary greatly and are sometimes even zero. Zero coverage rates can also be observed for $\hat{c}$ in some of the tested hyperprior settings. This implies that there are setups in which for all 100 replicates the true value for these two parameters never lies within the corresponding 95%-credible interval spanned by the quantiles of the empirical distribution formed by the posterior samples for `rho` or `sigma2`, respectively.
Potential reasons for this behavior should be investigated in more detail in further research by testing more combinations of hyperprior settings and focussing on `rho` or `sigma2`. For now, one can strongly suspect that these results are connected with an increased mean APE

and mean CRPS for certain combinations of hyperprior settings and test sample sizes `val_N` as depicted in Figure 8 and Figure 16.

Notably though, low coverage rates are not necessarily connected to higher mean APE and mean CRPS values. For example, Table 5 shows that the lowest average APE with a test set of size `val_N` = 500 is observed for Setting 4 and a training sample size of `N` = 250. In this case, coverage rates are 0.87 for $\hat{\mu}$, 0.89 for $\hat{c}$ and a coverage rate of zero is observed for the estimate of the spatial decay parameter. The highest average APE occurs for Setting 1.2 combined with a training sample size of `N` = 500 with coverage rates of 0.97 for $\hat{\mu}$ and coverage rates of zero for $\hat{c}$ and $\hat{\rho}$.
Setting 2 with `N` = 250 has the lowest average CRPS with `val_N` = 500. Coverage rates are 0.86 for $\hat{\mu}$, 0.90 for $\hat{c}$ and again a zero coverage rate for the estimate of the spatial decay parameter. Results for Setting 6.3 for `N` = 500 show the highest averaged CRPS evaluated with `val_N` = 500 where coverage rates are 0.97 for $\hat{\mu}$ and coverage rates of zero are observable for $\hat{c}$ and $\hat{\rho}$.

In general, the highest mean APE and mean CRPS values, i.e. the worst performance in spatial interpolation, are observed when coverage rates for $\hat{c}$ and $\hat{\rho}$ are both zero. If only estimates $\hat{\rho}$ show low coverage rates, the mean APE and mean CRPS values are not suggesting a considerably worse predictive performance compared to other setups where coverage rates for $\hat{\rho}$ are higher.
In fact, in the examples presented above, lowest values of mean APE and mean CRPS occur in cases where the coverage rate of $\hat{\rho}$ is zero while coverage rates for $\hat{\mu}$ and $\hat{c}$ are high. In these cases, the model is apparently able to perform a more accurate spatial interpolation than compared to the instances in which coverage rates of $\hat{c}$ and $\hat{\rho}$ are both non-zero and also compared to the cases where both true values of the two parameters are not included in any of the posterior samples for `rho` and `sigma2` in each of the 100 replicates.

3.6 Discussion

As illustrated in this section, wrapped Gaussian process models come with a complex modeling setup that requires the specification of different parameters like hyperparameters of assumed prior distributions that influence measures of prediction accuracy like APE and CRPS.
Here, an exploratory step was combined with an extended simulation setup for testing and

comparing different hyperparameter specifications. The focus is the comparison of different hyperprior settings for `alpha`, `sigma2` and `rho` as summarized in Table 1 and Table 3 as well as their interactions with different training and test sample sizes within the setup for wrapped Gaussian process models in `CircSpaceTime`. The presented simulated data examples show that, with large training sample sizes, hyperprior settings with an informative hyperprior setting for `rho` perform considerably worse than weakly informative settings as well as other informative settings with a broader range specified by the parameters of the Uniform prior for the spatial decay parameter `rho`.

It also appears that low coverage rates for modal point estimates $\hat{\rho}$ do not necessarily lead to a declining predictive performance compared to instances with higher coverage rates for the spatial decay parameter and instances in which both $\hat{c}$ and $\hat{\rho}$ show zero coverage rates. In particular, the best predictive performance on test data with sample size `N` $= 500$ is apparent when coverage rates for $\hat{c}$ are high while coverage rates for $\hat{\rho}$ are close or equal to zero. Investigating the interaction between `sigma2` and `rho` by testing more combinations of hyperparameter settings systematically is necessary for further exploring these effects.

As presented before, ρ together with σ^2 are the defining parameters of covariance matrices in the wrapped spatial Gaussian process. Thus, the spatial decay parameter ρ plays a crucial role in spatial interpolation.
Loosely speaking, the spatial decay parameter ρ controls how strongly the observed directions at neighboring points in space influence the predicted direction for a given point. Observing an increasing APE and CRPS for certain informative settings with a small range in the Uniform prior for `rho` might be connected with this central role of ρ.

A hypothesis of spatial overfitting can be formulated, explicitly, that for a certain size of training sample, like `N` $= 250$ or `N` $= 500$, a small range in the assumed Uniform prior for `rho`, defined by a small difference between the hyperparameters min and max, leads to an overemphasis on a rather close region around a given point in the space of training observations.
As there are a lot of observed directions in a large training set, it is easy to find a point where an observed direction is available from the training data in a small neighborhood around the given point of interest for which a direction should be interpolated. Combining this effect with an informative prior on `rho` in the model fitting step within `CircSpaceTime::WrapSp()` might cause a strong focus on a small neighborhood of points.

Thereby, the estimate for the spatial decay parameter obtained from a large training sample of e.g. `N` $= 500$ might not allow sufficient adaption needed for the spatial interpolation.
For small sample sizes, points are more sparsely distributed and, in order to find reference points with observed directions for spatial interpolation, a broader neighborhood has to be searched which gives less weight to whether the prior for `rho` was assuming a wide or small range in the model fitting step within `CircSpaceTime::WrapSp()` in the first place. Therefore, spatial interpolation on a test dataset is presumably not affected as much. Of course, such a hypothesis needs to be tested thoroughly in a more extended setup but at this point offers an interesting starting point for further discussion and research.

All in all, these findings illustrate that when applying wrapped spatial Gaussian process models, different selections of hyperparameter and hyperprior settings should be documented and tested. Results presented here suggest a potential trade-off between the inclusion of available knowledge via informative priors for the spatial decay parameter and potential spatial overfitting effects on large training samples.
Further research is needed to investigate predictive performance of wrapped Gaussian process models and the dependencies between different sample sizes and hyperprior settings (especially for the spatial decay parameter) and other influential factors e.g. via testing a systematic grid of hyperparameter combinations and by exploring a larger variety of different training and test sample sizes.

4 Modeling Wind Directions with Wrapped Gaussian processes

The following part shows an application of wrapped Gaussian process models for spatial interpolation of circular data using wind directions provided by the German Weather Service (Deutscher Wetterdienst (DWD)).
In particular, for a given set of wind directions at spatial locations in Germany, wrapped Gaussian process models are used for spatial interpolation of wind directions at unobserved locations. These real-data examples further illustrate model fitting and spatial prediction of circular spatial data using wrapped spatial Gaussian process models in `CircSpaceTime` (G. Jona Lasinio, Mastrantonio, and Santoro 2019).

4.1 Weather Data & Gaussian processes

Simply speaking, atmospheric phenomena like wind are part of a larger physical climate system that itself is connected to biogeochemical components shaping a joint system of interactions between the atmosphere, biogeochemical cycles and human life on earth. Via e.g. resource exploitation and emissions, human activities can influence the physical climate as well as the biogeochemical system and in turn, human activities are influenced by changes in climate, weather phenomena and ecosystems (see Latif (2009), p. 15).
As introduced in Gebhardt et al. (2020), climate consists of weather phenomena observed over time spans that are long enough to analyze statistical moments of atmospheric variables. On the other hand, weather denotes short-term atmospheric states.[6]
Wind is one of the atmospheric variables shaping observed weather phenomena. Hence, wind speed and wind direction are important components in models of the atmosphere.

In short, wind develops when different atmospheric pressure occurs at different spatial locations. These differences in atmospheric pressure typically result from and are influenced by intensities of other atmospheric variables, such as temperature or humidity (see chapter 8 in Gebhardt et al. (2020)).
Measuring wind essentially means measuring a wind vector at a given time and point in space. This can be done by directly measuring wind speed and wind direction, i.e. the length of the wind vector and its corresponding angle, or by measuring the meridional and zonal velocities of the wind vector (see online resources by Pidwirny and Jones (2010) for a detailed overview).

In a world facing consequences of anthropogenic climate change, analyzing and predicting weather phenomena is a subject of increasing importance. Latif (2009) emphasizes that observations of the last centuries suggest e.g. an overall increase of extreme weather events like droughts, floods or hurricanes. On these grounds, it is vital to monitor, model and predict wind directions.
Moreover, Toulkeridis and Zach (2017) present a case study where wind directions help in predicting distributions of volcanic ash for public and flight safety in Ecuador.
Also, the analysis of wind as a spatial phenomenon is a well-known illustrative example in directional statistics. Breckling (1989) analyzed wind directions in Australia and, as intro-

[6] As Latif (2009) points out, the atmosphere is an example for a chaotic system which naturally limits the predictability of weather phenomena to short time ranges.

duced earlier, Coles (1998) provides an example for the analysis of wind directions over time using a wrapping approach in a Bayesian framework.
Gneiting and Guttorp (2010) present spatio-temporal Gaussian processes including a case study for wind speed data in Ireland. Feng et al. (2018) applied Gaussian process regression for multivariate spatial interpolation of wind fields. Another example can be found in Lang et al. (2019) where the authors implement bivariate Gaussian models for predicting wind vectors making use of a distributional regression framework.
Also, Jona Lasinio, Santoro, and Mastrantonio (2020) provide case studies for the prediction of wind directions with wrapped and projected Gaussian process models illustrating approaches implemented in `CircSpaceTime` (G. Jona Lasinio, Mastrantonio, and Santoro 2019) with the use of wind direction measurements in Italy.

There are several reasons why Gaussian process models can be of interest when analyzing directional weather data. Firstly, weather data is typically available as time-series data e.g. measured over a spatial grid of measurement locations or as the output of meteorological models. Therefore, spatial and temporal dependence structures are present in weather data. As introduced in section 2, Gaussian process models can account for these spatial and temporal dependences by directly enabling inference in a space of functions (in space or time or both).
Also, section 2 shows that updating a Gaussian process model in a Bayesian framework comes down to updating posterior distributions over possible functions. Therefore, including new data in such a model is easy and transparent. Measurements of weather data like wind speed and wind direction are typically taken in regular time intervals (e.g. every 10 minutes) and as such there is a constant flow of incoming data available that can then be easily incorporated and used for updating a Gaussian process model.

When working with meteorological data it is important to note the difference between the meteorological wind direction and the mathematical notion of points on a unit circle which represent angles (that in turn represent directions).
The meteorological wind direction is typically defined as the horizontal direction from where the wind is blowing. For example, a southerly wind comes from the south and blows towards the north. The direction towards which the wind is blowing is called wind vector azimuth (cf. Pidwirny and Jones (2010)). For an illustration, Figure 9 juxtaposes two representations of a circular data vector $(\pi, \pi - 0.2, \pi - 0.4, 1.8\pi, 2\pi, \pi + 0.5)^T$ by depicting the values as points

on the unit circle on the left and as meteorological directions on the right-hand side. To transform one representation into the other, one has to rotate and flip the coordinate system.

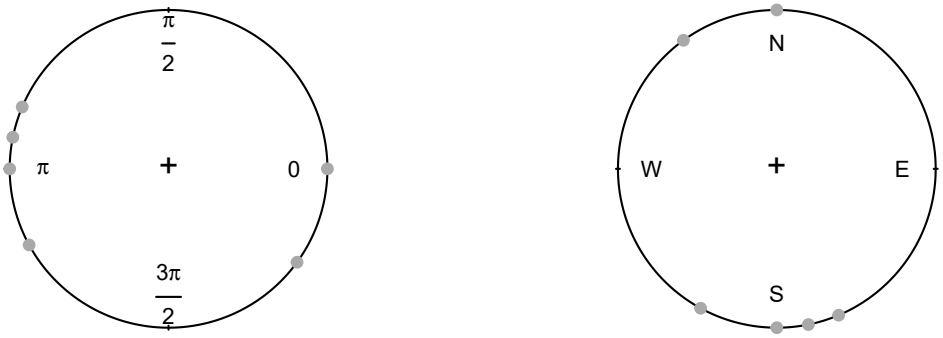


Figure 9: Directions as points on the unit circle and as meteorological directions. Directions in radians are $(\pi, \pi - 0.2, \pi - 0.4, 1.8\pi, 2\pi, \pi + 0.5)^T$ and plotted as grey dots on the circle to illustrate the two different representations.

4.2 DWD Dataset

The data used for the following analyses stems from the German Weather Service (Deutscher Wetterdienst, DWD) and is available as `.zip`-files via the DWD open data platform at `ftp://opendata.dwd.de`. Detailed steps of combining wind data and information on the respective weather stations can be seen in `download_data.R` which extends scripts by I. Marques & N. Umlauf.

The data used in the following examples contains 10-minute measurements of wind speed in $\frac{m}{s}$ (meter per second) and wind direction as angles measured in degrees averaged over the respective 10-minute intervals.
The given data provides dates and times of measurements as well as the location and additional information of the different weather stations in Germany. There are 266 weather stations available from which 251 provided wind direction measurements at all chosen time points.

In contrast to the wave direction data used in Jona Lasinio, Gelfand, and Jona Lasinio (2012) and G. Jona Lasinio, Gelfand, and Jona Lasinio (2019), the data at hand is not the output from a statistical forecast model over a symmetric spatial grid with fixed cell sizes but is available as measurements of wind directions at the different weather stations in Germany and given points in time. Jona Lasinio, Santoro, and Mastrantonio (2020) also analyze wind directions using observations from 50 km $\times$ 50 km grid cells that are transformed to provide

smaller grid cell sizes.
Different from these approaches, the following analyses directly work with the wind direction measurements taken at each weather station to illustrate results for observed directions in a non-equidistant grid of measurement locations.

4.2.1 Preprocessing

In the given dataset, wind direction is measured on a 360° scale and given as a meteorological wind direction, i.e. the direction from where the wind is blowing. Therefore, wind directions in degrees are transformed to radians via $2\pi \text{ rad} = 2\pi\frac{180°}{\pi} = 360°$.
This means that a value of $360° = 0° = 2\pi$ rad represents a northerly wind, i.e. wind blowing from the north, an easterly wind is observed when wind directions are $90° = \frac{\pi}{2}$ rad and a value of $180° = \pi$ rad corresponds to a southerly wind, i.e. wind blowing from the south and so forth.
Moreover, the coordinates of the weather stations are transformed from longitude and latitude to Universal Transverse Mercator (UTM) coordinates as the models in `CircSpaceTime` require locations in an UTM format.

Following approaches in G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) and Jona Lasinio, Santoro, and Mastrantonio (2020) who select wave or wind direction data at specific dates and points in time, the acquired data is filtered for four different dates of interest and the point in time is fixed to 10:00:00 AM for all selected dates as the focus is on spatial prediction here.
This selection of observation dates according to weather phenomena follows the approaches of G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) and Jona Lasinio, Santoro, and Mastrantonio (2020), where the authors compare spatial predictions of wave or wind directions obtained with wrapped Gaussian process models during stormy and calmer sea and weather states.
In particular, wind directions on 04.03.2019, 10.03.2019 in early spring and 24.06.2019 and 30.06.2019 in summer are taken into account. Haeseler et al. (2019) report that the two stormiest days of 2019 in Germany were storm "Bennet" on 04.03.2019 and storm "Eberhard" on 10.03.2019. To contrast these two days of strong storms with calmer periods, wind directions are also predicted on two days in June 2019 where a dominant high pressure area with low wind speeds and high temperatures was observed over Europe and Germany.
The motivation for this selection is to allow comparisons of spatial interpolation under dif-

ferent general tendencies and patterns of wind directions and wind speed at different points in the seasonal cycle.

Referring to ideas in G. Jona Lasinio, Gelfand, and Jona Lasinio (2019), where a day in summer with low wind speeds implied high variability of wind directions and lower interpolation accuracy while stormy weather showed similar wind directions with low variability and higher accuracy, it can be assumed that differences in the overall weather conditions like a strong storm or a heat wave at the selected points in time on 04.03.2019, 10.03.2019, 24.06.2019 and 30.06.2019 can lead to different prediction accuracy results as e.g. spatial prediction might be more challenging under high variability in wind directions.

4.3 Descriptive Insights

Figure 10 and Figure 11 show wind directions in Germany at the two different selected points in time in spring and summer, respectively. Wind vectors are pointing in the direction whereto

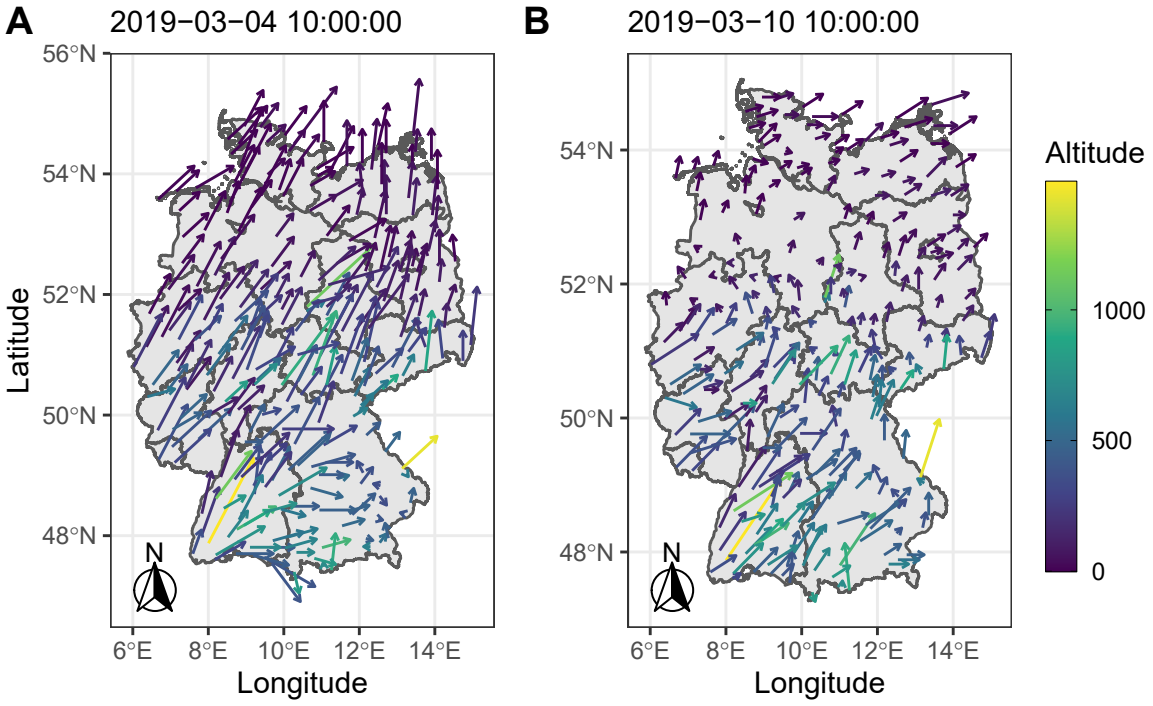


Figure 10: Maps with different wind directions observed at 10:00:00 AM on 04.03.2019 and 10.03.2019 in Germany (without Zugspitze). Wind speed measurements correspond to the length of the arrows. Altitudes of the given weather stations in meters above sea level are depicted with a continuous color scale.

the wind is blowing, i.e. the wind vector azimuth. The length of the arrows is proportional to wind speed measurements at the given weather station, i.e. high wind speed corresponds to a greater length of the arrow. Different altitudes of the given weather stations in meters

above sea level are depicted with a continuous color scale where lighter colors indicate higher altitudes. As expected for the selected dates, highest wind speeds are observed on 04.03.2019

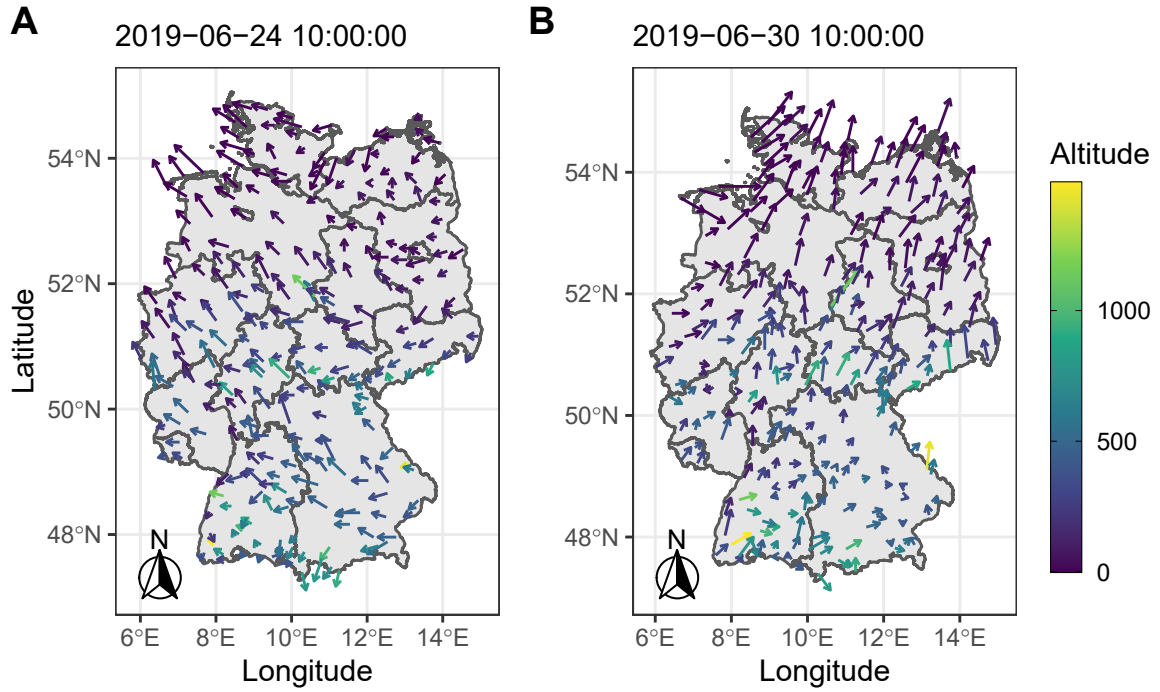


Figure 11: Maps with different wind directions observed at 10:00:00 AM on 24.06.2019 and 30.06.2019 in Germany (without Zugspitze). Wind speed measurements correspond to the length of the arrows. Altitudes of the given weather stations in meters above sea level are depicted with a continuous color scale.

and 10.03.2019, where wind directions follow joint patterns over large areas in Germany especially on 04.03.2019 (cf. Figure 10 (A)). Figure 10 (B) shows that, despite the general stormy weather state on that date, for the selected point in time on 10.03.2019 patterns of wind speed and wind direction measurements differ considerably between the southern and the northern parts of Germany. Figure 11 illustrates lower wind speed measurements on 24.06.2019 and 30.06.2019 with smaller and fewer areas of similar wind directions compared to Figure 10 (A).

Table 6 presents summary statistics of wind speed and wind direction at the selected dates and points in time. The circular mean of wind directions in radians is obtained with `circular::mean.circular()` from `circular` (Lund et al. 2017).
`CircStats::circ.disp()` from the R-package `CircStats` (Lund and Agostinelli 2018) returns circular dispersions of wind direction measurements on the selected dates. Circular standard deviation can be calculated with `circular::sd.circular()` using $\sqrt{-2\log(\bar{r}/n)}$, where $\bar{r}$ is the mean resultant length of the vectors of observed directions and n is denoting

the number of observations.

The wind direction patterns in Figure 10 show a rather homogenous wind field across Germany during the strong storm "Bennet" on 04.03.2019. This corresponds to the lowest circular dispersion of wind directions on 04.03.2019 (see Table 6). Higher variability in wind

	Wind Speed in m/s		Wind Direction in rad		
	mean	sd	circular mean	dispersion	circular sd
2019-03-04	9.81594	3.74746	3.948011	0.13447	0.53742
2019-03-10	6.11673	3.67150	3.953626	0.23022	0.72339
2019-06-24	4.55896	1.58496	1.750717	0.17190	0.61420
2019-06-30	4.52072	2.01900	3.886208	0.20270	0.67310

Table 6: Summary statistics of wind speed and wind direction measurements on 04.03.2019, 10.03.2019, 24.06.2019 and 30.06.2019 at 10.00.00 AM in the DWD dataset.

directions is observed for the selected point in time during storm "Eberhard" on 10.03.2019 as well as the two points in time in June. The highest circular dispersion can be observed for wind directions on 10.03.2019, possibly due to the fact that the storm was stronger in the southern part of Germany than in the northern parts for the selected point in time (cf. Figure 10 (B)).

The two selected points in time in June 2019 show larger variability than on 04.03.2019 but lower circular dispersion compared to the conditions on 10.03.2019. Wind directions vary slightly less on 24.06.2019 than on 30.06.2019.

In accordance with visual impressions in Figure 10 and Figure 11, mean wind directions on 04.03.2019, 10.03.2019 and 30.06.2019 in Table 6 are very similar, indicating an overall westerly wind at these dates and points in time. The rose diagrams for the selected days and time points in Figure 12 further illustrate these conditions by providing a circular equivalent of histograms for wind directions using `CircSpaceTime::rose_diag()`.

On the 04.03.2019, 10.03.2019 and 30.06.2019 wind is predominantly blowing from westerly directions, whereas on 24.06.2019 the wind direction differs considerably from the other dates as winds over Germany were dominantly blowing from easterly directions on that date and point in time.

4.4 Wrapped Spatial Model for Wind directions

Using wrapped Gaussian process models and the setup implemented in the `R`-package `CircSpaceTime`, this part explores spatial interpolation of wind directions. In particular, wrapped Gaussian

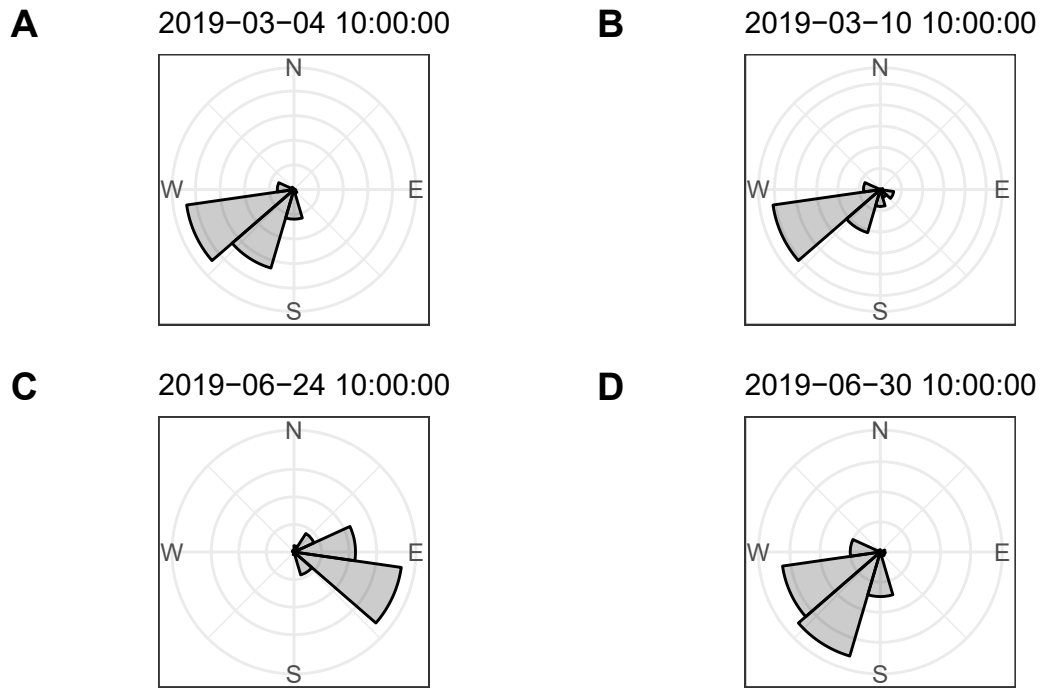


Figure 12: Circular equivalent of histograms for wind directions on 04.03.2019, 10.03.2019, 24.06.2019 and 30.06.2019 in Germany.

process models for the different selected dates produce sets of posterior samples for the model parameters that are then used to predict wind directions on a test dataset. Firstly, results for two differents sets of training and test locations are presented. Then, 10-fold cross-validation is used to compare results between dates and different hyperprior settings.

4.4.1 Spatial interpolation at two sets of test locations

There are 251 weather stations with data available on all selected dates and points in time. As a first step, two different sets of test locations are sampled from all available weather stations each with `dplyr::sample_frac(0.2)` from the `R`-package `dplyr` (Wickham et al. 2020). This results in two different splits each with 201 locations used for training and 50 locations left out for spatial interpolation. Figure 13 shows the two different selected sets of test locations for the evaluation of the models as black dots and respective training locations as grey dots. To obtain spatial predictions, two chains are run in parallel with `CircSpaceTime::WrapSp()` each creating 1500 samples for estimation with $30,000$ MCMC iterations with a burn-in of $15,000$ and a thinning of 10.

As in Chapter 3, an exponential kernel function is selected. Hyperparameter assumptions in all models are $\mathcal{WN}(\pi, 10)$ for the mean `alpha`, $\text{Inv}\Gamma(3, 0.5)$ for the variance `sigma2` and

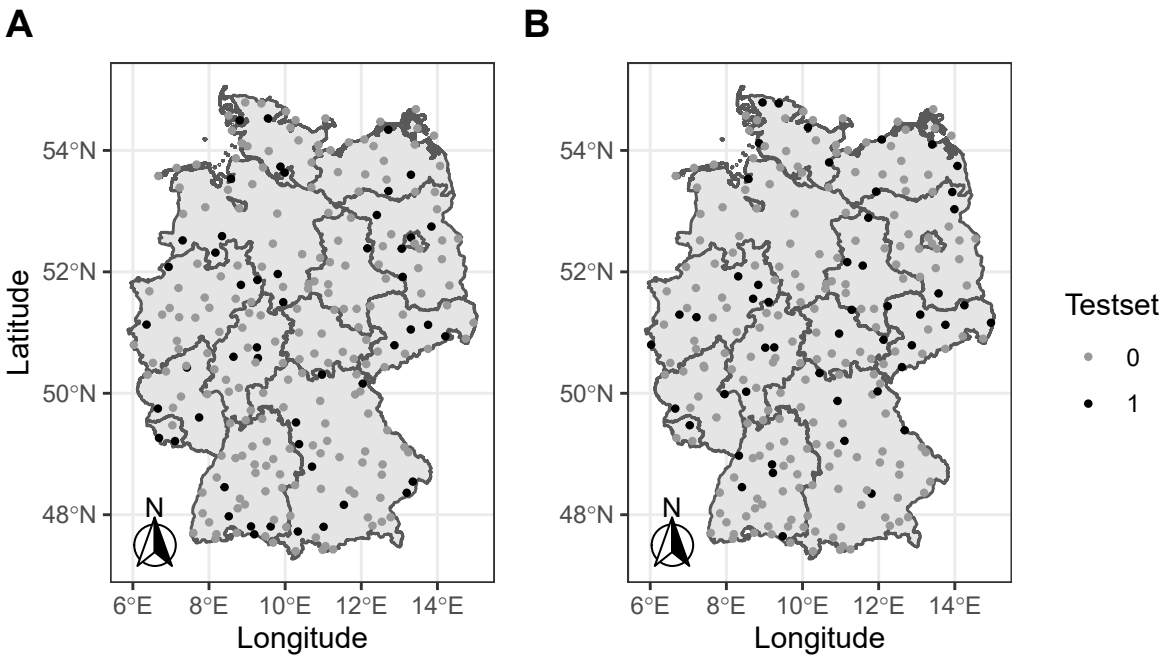


Figure 13: Two different sets of test locations as black dots, locations for training depicted as grey dots.

a Uniform distribution with maximum and minimum calculated from the distance matrix of the spatial locations for the spatial decay parameter `rho` following examples provided in the vignette of (G. Jona Lasinio, Mastrantonio, and Santoro 2019) and the analysis of wind directions in Jona Lasinio, Santoro, and Mastrantonio (2020).

In particular, the minimum of `rho`'s Uniform prior is defined anti-proportional to the maximum distance of all distances between all locations in the wind direction data and the value chosen for the maximum is obtained anti-proportional to the minimum distance between locations (cf. Jona Lasinio, Santoro, and Mastrantonio (2020)).

Referring to Jona Lasinio, Gelfand, and Jona Lasinio (2012), the decay parameter and the variance update jointly via an adaptive Metropolis procedure starting with initial guesses `sd_prop=list("sigma2"=1,"rho"=0.3)` for both parameters.

All steps required for fitting and evaluating the model as well as for spatial predictions are defined as functions in `twomodels_functions.R` that are then applied to a nested dataframe containing data on the four selected dates (see `fit_twomodels.R` for the complete steps).

Predictions are obtained using `purrr::map()` from `purrr` (Henry and Wickham 2020) combined with a custom defined function `get_split_fit_pred()` in `twomodels_functions.R` that takes as input a given dataset and a vector of test locations and returns a list column of pre-

dictions. Values for the potential scale reduction factor $\hat{R}$ as well as for mean APE, baseline APE and mean CRPS for each model at each of the four selected dates are calculated from the nested results using `purrr::map_dbl()` from `purrr` (Henry and Wickham 2020) and the defined functions in `twomodels_functions.R`, like `get_meanape()` or `get_meancrps()`.

For two or more chains in `CircSpaceTime::WrapSp()`, convergence of the results can be evaluated with `CircSpaceTime::ConvCheck()` which allows the creation diagnostic plots and returns potential scale reduction factors ($\hat{R}$s) of the model parameters and a multivariate Potential Scale Reduction Factor for combinations of variables in multivariate chains using `coda::gelman.diag()` from the `R`-package `coda` (Plummer et al. 2020). Following the documentation of `coda` (Plummer et al. 2020), convergence can be assumed when the upper limit of $\hat{R}$ is close to 1. Results for $\hat{R}$, its upper confidence limit and the multivariate $\hat{R}$ for each of the models are presented in Table 7 and suggest convergence for all model parameters. For more details on the convergence dynamics of the models presented here, traceplots[7] and

	`alpha`		`sigma2`		`rho`		
Obs Date	$\hat{R}$	upper CI	$\hat{R}$	upper CI	$\hat{R}$	upper CI	multivar. $\hat{R}$
Locations 1							
2019-03-04	0.9998850	1.0000245	1.0023670	1.011925	1.0529086	1.0903251	1.0064784
2019-03-10	1.0001209	1.0001655	1.0245245	1.053542	1.0076243	1.0257414	1.0056996
2019-06-24	0.9996720	0.9996929	1.0011134	1.002516	1.0014779	1.0031561	1.0000606
2019-06-30	1.0021584	1.0120168	0.9999067	1.000797	1.0016102	1.0084495	1.0043552
Locations 2							
2019-03-04	0.9999653	1.0005376	0.9997373	1.000017	0.9997541	0.9998929	0.9998426
2019-03-10	1.0024724	1.0026515	1.0071842	1.029994	1.0101318	1.0336037	1.0053480
2019-06-24	1.0005135	1.0031283	1.0019748	1.008804	1.0063095	1.0136966	1.0025402
2019-06-30	0.9998527	1.0005885	1.0011911	1.007142	1.0080992	1.0172627	1.0041444

Table 7: $\hat{R}$ and its upper CI for each model parameter and multivariate $\hat{R}$ for all models on 04.03.2019, 10.03.2019, 24.06.2019 and 30.06.2019. Upper limit of $\hat{R}$ close to 1 means approx. convergence has been reached.

plots of running means are created with the `R`-package `ggmcmc` (Fernández i Marín 2020) and can be found in Figures 19, 20, 21, 22 as well as Figures 23, 24, 25 and 26 in the Appendix. For example, Figure 21 (B) and Figure 20 (A) and (B) show a great variety in the two chains for `rho`, whereas the chains for `rho` in Figure 19 (A) as well as for both different training datasets in Figure 22 are very concentrated with a few large peaks.

Plots of running means show how quickly parameter values approach the respective mean over all iterations (cf. Fernández i Marín (2020)). Figure 24 (B) shows an example of parameter

[7] For the interpretation of traceplots for `alpha`, the circular nature of `alpha` has to be considered (cf. Jona Lasinio, Santoro, and Mastrantonio (2020))

values of `rho` and `sigma2` varying up until higher interations of the respective MCMC chains. Figure 23(A), Figure 24 (A) and Figure 26 (A) illustrate that for the first training dataset and observations on 04.03.2019, 10.03.2019 and 30.06.2019, parameter values approached their respective mean faster than for the second training dataset.

Figure 14 and Figure 15 show predicted wind directions for the two different sets of training and test data on 10.03.2019 and 30.06.2019, respectively. Predicted wind directions are depicted as bold black arrows of equal length for the different models and the two different sets of test locations. For both dates and the two different sets of test locations, there are

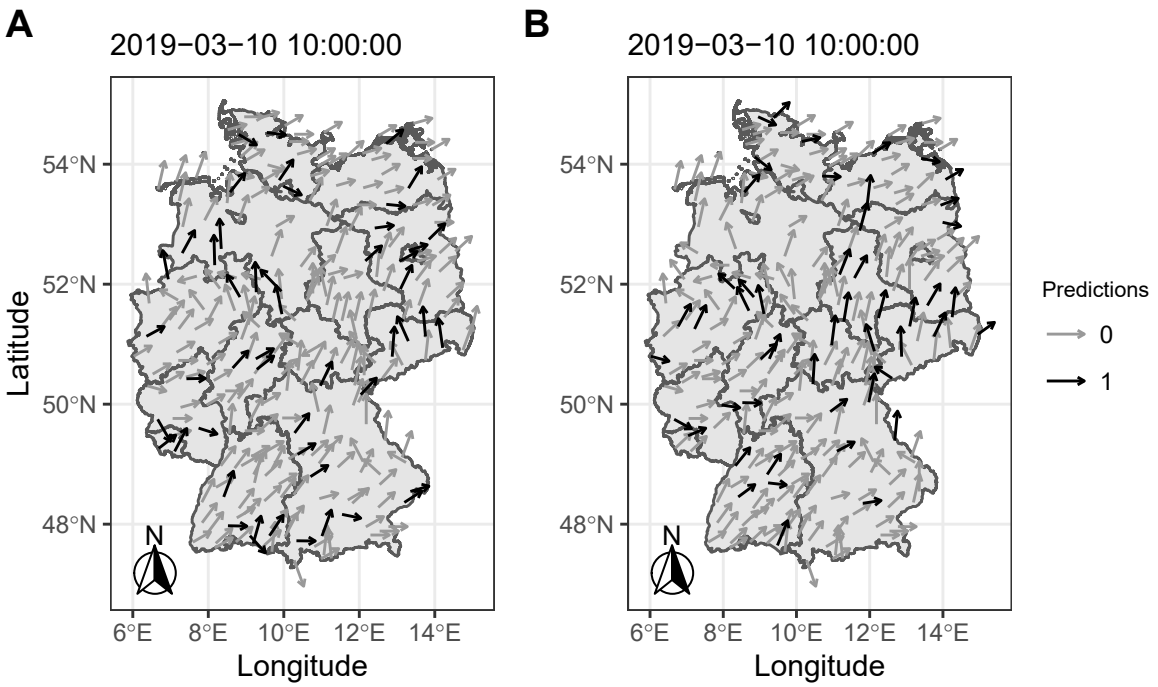


Figure 14: Predicted wind directions (black arrows) at first (A) and second (B) set of test locations for data observed at 10:00:00 AM on 10.03.2019. Training observations depicted as grey arrows.

some areas where the predicted direction at a given test location follows the general pattern observed around the given location and other areas where the predictions and the training observation differ. All in all, there is a lot of variability in observed and predicted directions and this illustration is of course limiting the attention to some of the more obvious patterns of predicted and observed wind directions. Similar plots illustrating predicted wind directions for 04.03.2019 and 24.06.2019 can be found in the Appendix in Figure 27 and Figure 28, respectively.

For model evaluation, circular variants of the average prediction error (APE) and the contin-

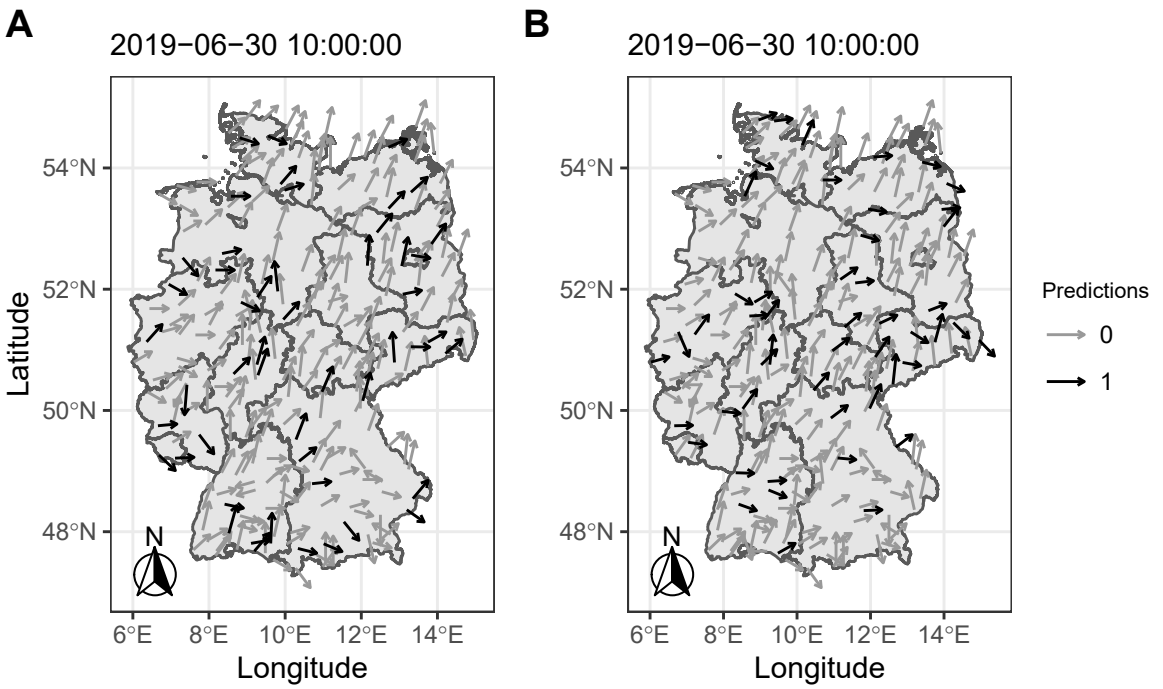


Figure 15: Predicted wind directions (black arrows) at first (A) and second (B) set of test locations for data observed at 10:00:00 AM on 30.06.2019. Training observations depicted as grey arrows.

uous ranked probability score (CRPS) are calculated for the observations not used for model fitting for each selected date and the two different sets of test locations.[8] Table 8 shows the results for mean APE und mean CRPS of the four different models and two different sets of test locations as well as mean APE values for a baseline model that uses mean wind direction in the respective training data as predictions.

Over all four selected dates, results of spatial interpolation on 10.03.2019 show the lowest mean APE and mean CRPS for both sets of test locations, i.e. the highest prediction accuracy among all selected dates as measured by mean APE and mean CRPS. At the same time, Table 6 depicts the highest variability in wind directions for the selected point in time on 10.03.2019. The highest mean APE and mean CRPS and therefore the lowest prediction accuracy are observed for both sets of test locations on 30.06.2019, where wind directions have the second highest circular variability in Table 6. The second highest mean APE and mean CRPS values for both test locations occur on 04.03.2019, where Table 6 shows the lowest variability in wind directions.

[8] As introduced in Chapter 3, a circular version of the APE can be calculated with `CircSpaceTime::APEcirc()`. CRPS can be used to compare the performance of the predictive distribution with the circular observations in the test set at a given location using `CircSpaceTime::CRPScirc()`.

Obs Date	mean APE	mean APE (baseline)	mean CRPS
Locations 1			
2019-03-04	0.3734612	0.1353413	0.2403547
2019-03-10	0.1787378	0.2642929	0.0654557
2019-06-24	0.2722876	0.1471928	0.1374098
2019-06-30	0.4904246	0.2362895	0.3099265
Locations 2			
2019-03-04	0.3673710	0.0982334	0.2321860
2019-03-10	0.1950385	0.2319723	0.0965984
2019-06-24	0.2584875	0.1526705	0.1353099
2019-06-30	0.5081791	0.1754892	0.3098312

Table 8: Mean APE and mean CRPS of wrapped Gaussian process models and mean APE of baseline model that uses mean wind direction in training data as predictions. Results are shown for the four selected dates and the two different sets of test locations.

These results are not in accordance with the idea formulated earlier that high variability in wind directions, like on 10.03.2019 (cf. Table 6), could lead to more uncertainty in spatial predictions. In G. Jona Lasinio, Gelfand, and Jona Lasinio (2019), the authors report lower prediction accuracy for wave directions in calmer sea states where increased variability of circular measurements is observed (cf. G. Jona Lasinio, Gelfand, and Jona Lasinio (2019), p.138). Here, the connection between weather states in spring and summer, different variability in wind directions and prediction uncertainty is not as clear.

Comparing the results for the two different sets of test locations in Table 8, slightly lower values of mean APE and CRPS are observed for the second set of test locations on 04.03.2019 and on 24.06.2019, whereas for 10.03.2019 and 30.06.2019 higher values occur for the second set of test locations. Overall, values of mean APE and mean CRPS differ more between the different selected dates than between the selected two different test locations.

As presented in Chapter 3 for the simulated examples, Table 8 also shows results for a baseline model where wind directions at unobserved locations are predicted with the mean wind direction in the given training dataset.
Interestingly, in contrast to results in Table 2 and Table 4 in Chapter 3 where all models performed worse than this baseline, the results in Table 8 show that there are in fact setups in which the wrapped Gaussian process models show a higher prediction accuracy compared

to the baseline, even though for 04.03.2019, 24.06.2019 and 30.06.2019 mean APE values are again lower for the baseline model.
In particular, prediction accuracy is higher than the baseline for the wrapped Gaussian model on 10.03.2019, i.e. on a stormy day with a high circular variability of wind directions across the weather stations in Germany as summarized in Table 6.
This suggests that wrapped Gaussian process models might not necessarily perform better than predicting wind directions at unobserved locations with mean wind direction when circular variability is low, i.e. in weather states with patterns of wind directions that are similar across all weather stations as e.g. depicted for 04.03.2019 in Figure 10 (A). On the other hand, results for 10.03.2019 in Table 8 indicate that wrapped Gaussian process models can provide an improvement in spatial interpolation compared to the baseline model when wind directions show higher variability between the given measurement locations.

4.4.2 Testing hyperprior settings with 10-fold cross-validation

For comparing different hyperprior settings, cross-validation methods enable evaluating the above models without limiting the perspective to specifically selected sets of test locations, thus controlling for the influence of one specific set of test locations on the accuracy of spatial predictions. Therefore, models for the four selected dates are refitted and evaluated using k-fold cross validation with $k = 10$ for three different hyperprior settings in `CircSpaceTime::WrapSp()`.

The first selected hyperprior setting is the same setting as in the models presented in section 4.4.1 for two sets of test locations. As described in section 4.4.1, the first setting assumes a weakly informative setup with $\mathcal{WN}(\pi, 10)$ for `alpha`, Inv$\Gamma(3, 0.5)$ for the variance `sigma2` and a Uniform distribution with minimum and maximum chosen according to the distances between the spatial locations where the minimum and the maximum of `rho`'s Uniform prior are defined anti-proportional to the maximum and minimum distance, respectively.
The second setting corresponds to Setting 2 from section 3, i.e. $\mathcal{WN}(\pi/2, 2\pi)$ for `alpha`, a weakly informative setting with $\mathcal{U}(0, 5)$ for `rho` and Inv$\Gamma(0.1, 0.1)$ for `sigma2`.
The third setting assumes also $\mathcal{WN}(\pi/2, 2\pi)$ for `alpha` and Inv$\Gamma(0.1, 0.1)$ for `sigma2` but another range in the Uniform prior for the spatial decay parameter `rho` with $\mathcal{U}(0.001, (\texttt{rho}_{max}/2)+ 0.02)$ where $\texttt{rho}_{max}$ is caluclated anti-proportional to the minimum distance in the distance matrix of spatial locations.

Details of running 10-fold cross-validation models can be found in `10cv_functions.R` and `10cv_run_create_tables.R`. Cross-validation splits for the four selected dates are obtained using `rsample::vfold_cv()` from the `R`-package `rsample` (Kuhn, Chow, and Wickham 2020). Again, `purrr::map_dbl()` from `purrr` (Henry and Wickham 2020) and the defined functions in `10cv_functions.R` are used to get cross-validated multivariate $\hat{R}$, APE, baseline APE and CRPS values at each of the four selected dates and for the three different hyperprior settings.

Multivariate $\hat{R}$ values in Table 9 averaged over the cross-validation runs suggest convergence for the models as values are ≥ 1. The results for mean APE and CRPS as well as mean APE

Obs Date	multivar. $\hat{R}$
Setting 1: min, max	
2019-03-04	1.001989
2019-03-10	1.015526
2019-06-24	1.006791
2019-06-30	1.164215
Setting 2: from Simulation	
2019-03-04	1.002125
2019-03-10	1.021942
2019-06-24	1.006270
2019-06-30	1.199763
Setting 3: small range	
2019-03-04	1.002179
2019-03-10	1.009809
2019-06-24	1.007417
2019-06-30	1.039378

Table 9: 10-fold cross-validation mean multivariate $\hat{R}$ for all combinations of three different hyperprior settings and four different datasets from 04.03.2019, 10.03.2019, 24.06.2019 and 30.06.2019.

for a baseline model (averaged over all 10 folds) are presented in Table 10.

Similar to model evaluation results for two different test locations from earlier, cross-validated results for the prediction of wind directions on 10.03.2019 show the lowest APE and CRPS values over all three hyperprior settings compared to the other selected dates. Again, the highest values for mean APE and mean CRPS are observed for spatial predictions on 30.06.2019 over all three hyperprior settings.

In contrast to results for two sets of test locations, cross-validated mean APE and CRPS

Obs Date	mean APE	mean APE (baseline)	mean CRPS
Setting 1: min, max			
2019-03-04	0.2466659	0.1350021	0.1400314
2019-03-10	0.1851788	0.2312078	0.0947648
2019-06-24	0.2654181	0.1737227	0.1473190
2019-06-30	0.4507136	0.2035279	0.2599462
Setting 2: from Simulation			
2019-03-04	0.2428762	0.1350021	0.1351850
2019-03-10	0.1852742	0.2312078	0.0942933
2019-06-24	0.2609799	0.1737227	0.1384835
2019-06-30	0.4395015	0.2035279	0.2778178
Setting 3: small range			
2019-03-04	0.2439925	0.1350021	0.1352940
2019-03-10	0.1853868	0.2312078	0.0939885
2019-06-24	0.2591190	0.1737227	0.1392187
2019-06-30	0.4510274	0.2035279	0.2530700

Table 10: 10-fold cross-validation mean APE, mean CRPS and mean APE for a baseline model for all combinations of three different hyperprior settings and four different datasets from 04.03.2019, 10.03.2019, 24.06.2019 and 30.06.2019. The baseline model predicts wind direction at unobserved locations with the overall mean wind direction in the given training dataset.

values for 04.03.2019 and 24.06.2019 only differ marginally and Table 10 shows slightly lower cross-validated APE and CRPS values on 04.03.2019 than on 24.06.2019. Overall, 10-fold cross-validation shows only small variations between the three different hyperprior settings for all four selected points in time. The differences in cross-validated mean APE and CRPS values between the different dates outweigh the differences between the three selected hyperprior settings.

In contrast to results for two sets of test locations, the lowest circular dispersion on 04.03.2019 in Table 6 corresponds to the second lowest cross-validated mean APE and CRPS values while slightly higher values occur on 24.06.2019, where the second lowest variability in wind directions is presented in Table 6. In accordance with results for two sets of test locations, the lowest cross-validated APE and CRPS, i.e. the highest prediction accuracy, is observed on 10.03.2019 where also the largest variability in wind directions is observed (cf. Table 6).

Jona Lasinio, Santoro, and Mastrantonio (2020) report higher APE and CRPS for wrapped

Gaussian process models on a selected day in summer with large variability in wind directions and lower APE and CRPS for a selected stormy day in autumn where winds have more similar directions. G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) observe that lower variances in the directional observations in stormy sea states correspond to higher prediction accuracy and lower predcition accuracy is present for calm sea states with higher variation of wave directions in their analyses.
Referring to the summary in Table 6 and results in Table 10, it is evident that in the cross-validated setup presented here no clear pattern like in Jona Lasinio, Santoro, and Mastrantonio (2020) or G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) is detected. All in all, cross-validated results support the findings for two sets of test locations in section 4.4.1 that the connection between dispersion of directional observations and the prediction accuracy measured by mean APE and mean CRPS is ambiguous for the presented examples of wind directions.

Results for cross-validated APE of a baseline model in Table 10 are similar to the results for two sets of test locations (cf. Table 8). Once more, mean wind direction in the respective training set are used as predictions in the baseline model. Following results for two sets of test locations, cross-validated APE values of spatial interpolation for 04.03.2019, 24.06.2019 and 30.06.2019 are lower in the baseline model than for the wrapped Gaussian process models (cf. Table 10), i.e. in these cases predictive performance of wrapped Gaussian process models is worse than spatial interpolation accuracy obtained with the baseline model.
Notably, for 10.03.2019, the cross-validated APE is lower for the wrapped Gaussian model than for the baseline model indicating that at this selected date using mean wind direction as predictions at the unobserved locations leads to higher errors than using predictions provided by the wrapped Gaussian process model.

These cross-validated results further support the hypothesis formulated earlier that wrapped Gaussian process models provide better prediction accuracy than using mean wind directions when variability between the observed circular measurements is high (like on 10.03.2019), while performing worse when circular dispersion is lower, i.e. when wind directions are mostly the same as e.g. observed on 04.03.2019.
Future research should compare results for different baselines and wrapped Gaussian process models for a larger number of dates and points in time with different levels of circular dispersion to investigate these results further.

4.5 Summary and Limitations

The above examples for wind directions in Germany illustrate applications of wrapped Gaussian processes for the analysis of circular data. For two different sets of test locations, mean APE and CRPS values differ more between selected dates than between different test locations. Results from 10-fold cross-validation suggest that prediction accuracy of wrapped Gaussian process models measured by APE and CRPS differs more widely between different selected dates than between different hyperprior settings in `CircSpaceTime::WrapSp()`.

In contrast to G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) and Jona Lasinio, Santoro, and Mastrantonio (2020), the connection between stormy and calm weather states, different levels of variability in observed directions and the prediction accuracy as measured by APE and CRPS is ambiguous for the presented examples of spatial interpolation of wind directions with wrapped Gaussian process models.

Wrapped Gaussian process models for wind directions mostly perform worse than assuming the mean wind direction at the unobserved locations but predictions of the wrapped Gaussian process model for 10.03.2019 show higher accuracy, i.e. lower mean APE, than the baseline model. This suggests that for high variability of wind directions at the selected date, wrapped Gaussian process models can provide an improvement in spatial interpolation compared to a baseline model assuming the mean wind direction at all unobserved locations.

Still, it is important to emphasize that the models presented here serve an illustrative purpose but are most likely oversimplifying and thus improveable. In contrast to e.g. Jona Lasinio, Gelfand, and Jona Lasinio (2012), G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) and Jona Lasinio, Santoro, and Mastrantonio (2020), the presented analyses use data directly measured at the given weather station that is therefore not distributed on an equidistant grid. Instead, spatial distances between wind direction measurements and altitudes of the measurement locations vary.
Moreover, observed wind directions are not only connected to different altitudes of the weather stations but also are potentially dependent on other environmental variables specific to the given location.
It was not in the range of this work to assess the possible impact of asymmetric spatial grids and other factors like altitude differences and it can therefore only be assumed that these effects additionally influence spatial interpolation.

As presented in Chapter 3, prediction accuracy of spatial interpolation with wrapped Gaussian processes is influenced by specific combinations of training sample sizes and hyperprior settings set in `CircSpaceTime::WrapSp()`. In this chapter, selected test locations are varied and three different hyperprior settings are compared in a 10-fold cross validated setup for four different dates in spring and summer. In the presented examples, spatial interpolation accuracy differs more between different selected dates than between different test locations or hyperprior settings. To investigate these findings further, future analysis of wind directions or other real-data applications using wrapped Gaussian process models should extend the evaluation of different hyperparameter and hyperprior specifications while comparing a wider range of different sets of test locations or making use of cross-validation methods.

5 Conclusions

This work explored wrapped spatial Gaussian process models developed by Jona Lasinio, Gelfand, and Jona Lasinio (2012) by investigating hyperprior sensitivity with simulated data and by providing examples for modeling and interpolating wind directions with wrapped Gaussian process models implemented in `CircSpaceTime` (G. Jona Lasinio, Mastrantonio, and Santoro 2019).

In section 3, wrapped spatial Gaussian process models were evaluated in simulation setups for 6 and 14 different hyperparameter settings and different training and test sample sizes using 100 replicates for each combination in section3.5.1 and section 3.5.2.
Additionally, simulated results were compared to a baseline model in Table 2 and Table 4. Results for a baseline model illustrated that, overall, models in the simulated examples performed worse than assuming the mean wind direction as a prediction at the unobserved locations.

Comparisons of different hyperparameter settings in the prior distributions of the model parameters showed that certain combinations of hyperparameter settings (especially for the Uniform prior of the spatial decay parameter) and training sample sizes led to increasing APE and CRPS for larger training data, i.e. decreasing predictive performance.
In particular, decreasing spatial interpolation accuracy was observed for combining larger training sample sizes like `N` = 500 with an informative hyperprior setting for Uniform prior of the spatial decay parameter `rho` compared to the mean APE and CRPS values in weakly informative settings as well as other informative settings with a broader range specified by

the minimum and maximum of `rho`'s Uniform prior (cf. Figure 6, Figure 7 in section3.5.1, Figure 8 in section 3.5.2 and Figure 16 in the Appendix).

Investigating mean point estimates averaged over replicates in each combination of hyperprior settings and training and test sample sizes in Table 5 showed that low coverage rates for averaged modal point estimates $\hat{\rho}$ not necessarily corresponded to worse predcitive performance. Only when low coverage rates were observed for $\hat{\rho}$ and $\hat{c}$ (where $c = e^{-\sigma^2/2}$) and training sample sizes were large, predictive accuracy of spatial interpolation was considerably worse than for the other investigated models.
These results emphasize the importance of hyperparameter settings for the prior distributions of the spatial decay parameter ρ and the jointly updated variance σ^2 for spatial interpolation with wrapped Gaussian process models.

Moreover, a hypothesis of spatial overfitting can be formulated as in section 3.6, i.e. results presented in Table 5, Figure 8 and Figure 16 suggest that for specific combinations of training sample sizes and an informative prior on `rho`, the resulting posterior estimation might overemphasize the available information in the training data with the help of an informative prior for `rho` and thereby the model generalizes worse to unobserved locations than for smaller training sample sizes.
Testing different combinations of hyperparameter settings especially for `rho` and `sigma2` as summarized in Table 3, the results presented here illustrate that these effects occur rather independent from the hyperparameter settings for `alpha` and `sigma2`, underlining the central role of `rho` and its hyperprior specifications for spatial interpolation.

Thus, when working with wrapped spatial Gaussian process models, such potential spatial overfitting effects should be taken into consideration. As illustrated here, a trade-off might occur between the inclusion of available knowledge via large training datasets combined with informative hyperparameter settings for the prior distribution of the spatial decay parameter and the ability of a wrapped Gaussian process model to generalize and interpolate directions at previously unobserved locations.
For now, these ideas remain to be investigated thoroughly in future research by e.g. testing a higher variety of hyperparameter and hyperprior settings for `rho` and other combinations of settings systematically to further explore predictive performance for certain combinations of hyperprior settings and training and test sample sizes.

Notably, differing from the approaches in e.g. Jona Lasinio, Gelfand, and Jona Lasinio (2012), G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) and Jona Lasinio, Santoro, and Mastrantonio (2020), locations in the simulated examples in section 3 were sampled from a grid of points leading to asymmetric distances between the selected points for training and testing. It was not in the range of this work to evaluate different spatial grids but it can only be assumed that the predictive accuracy differs between different grid specifications.

Providing a real-data application, section 4 focused on spatial interpolation of wind directions in Germany with a dataset from the German Weather Service (DWD). Different sets of test locations as well as three different hyperior settings for the wrapped Gaussian process models were compared.
Exploring prediction accuracy for two different sets of test locations, results for mean APE and mean CRPS differed less between the different selected locations than between the different dates (cf. Table 8). Moreover, three different hyperprior settings were compared using 10-fold cross-validation. Cross-validated mean APE and mean CRPS differed less between the different hyperprior setting than between the different selected dates (cf. Table 10).
In summary, these findings suggest that the selected dates (and points in time) and their specific characteristics, like differing variability in wind directions, have a strong impact on prediction accuracy.

Summary statistics in Table 6 and results in Table 8 imply that different levels of circular dispersion are connected to differing prediction accuracy as assessed by mean APE and mean CRPS values. In contrast to patterns suggested in G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) and Jona Lasinio, Santoro, and Mastrantonio (2020), highest prediction accuracy is observed for the selected date with the highest variability in wind directions on a stormy day (cf. Table 6, Table 8 and Table 10).

The potential impact of circular dispersion on prediction accuracy is further emphasized by comparing mean cross-validated APE of wrapped Gaussian process models to a baseline model that used mean wind direction of the respective training dataset as predictions at the unobserved locations (cf. Table 10).
Wrapped Gaussian process models for wind directions mostly performed worse than assuming the mean wind direction at the unobserved locations but predictions of the wrapped Gaussian process model for 10.03.2019 showed higher accuracy, i.e. lower mean APE, than the baseline model. Thereby, these cross-validated results suggest that, for high variability in

wind directions at the selected date, wrapped Gaussian process models provide an improvement in spatial interpolation accuracy compared to a baseline model assuming the mean wind direction at all unobserved locations.

As discussed in section 4.5, the results for spatial interpolation of wind directions in this work have specific limits. For example, dependencies on wind speed and other important atmospheric and geographic variables specific to the selected point in time and test location like temperature or altitude are not accounted for.
Moreover, in contrast to e.g. G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) or Jona Lasinio, Santoro, and Mastrantonio (2020), wind direction measurements are analyzed directly at the location of each of the 251 weather stations in Germany, thus also assuming an asymmetric and non-equidistant grid of measurement locations. Investigating the potential effects of different grid specifications on the interpolation of e.g. wind directions remains a task for future exploration of spatial interpolation with wrapped Gaussian processes.

As pointed out in section 2, there are a lot of extensions to the wrapped spatial Gaussian model available such as the wrapped skewed Gaussian process and the spatio-temporal Gaussian process as presented e.g. in Mastrantonio, Gelfand, and Jona Lasinio (2016), Mastrantonio, Jona Lasinio, and Gelfand (2016) and G. Jona Lasinio, Gelfand, and Jona Lasinio (2019), the latter also presenting a way of jointly modeling wave heights and wave directions. A similar approach could be applied for joint modeling of wind speed and wind directions. Thus the aforementioned extensions could enable more realistic ways of analyzing wind directions e.g. in combination with wind speed measurements.
Moreover, a wrapped spatio-temporal Gaussian process model as well as a projected Gaussian process model for the spatial and spatio-temporal setup are also implemented in `CircSpaceTime` (G. Jona Lasinio, Mastrantonio, and Santoro 2019) and could be tested for the prediction of wind directions over points in space and time in future work.

Overall, this work provides insights on spatial interpolation with wrapped Gaussian process models and starting from the presented ideas and results, there are several interesting topics to follow.
When applying wrapped Gaussian process models, special care should be taken to carefully investigate predictive performance under different combinations of informative hyperparameter settings for the spatial decay parameter `rho` in `CircSpaceTime` (G. Jona Lasinio, Mastrantonio, and Santoro 2019) and large training sample sizes.

For real-data examples, Jona Lasinio, Santoro, and Mastrantonio (2020) illustrate how maximum and minimum of `rho`'s Uniform prior can be chosen with respect to the distances between the given locations which might provide orientation for choosing hyperparameters in future applications.
In general, it is advisable to compare different choices of hyperparameters for the wrapped Gaussian process model for smaller and larger training sample sizes exploring whether certain combinations of hyperpriors and training and test sample sizes show an increasing or decreasing predictive performance in the given datasets or simulated setup.

For applications of wrapped Gaussian process models to real data, comparison to a baseline model for wind directions illustrated that, with higher variability in the observed directions, wrapped Gaussian process models and their flexibility can provide higher prediction accuracy. To investigate these findings further, future analyses should compare wrapped Gaussian process models to different baseline models on a larger number of different dates and points in time with differing characteristics, e.g. different degrees of circular dispersion and different general weather states like storms, calmer weather and transition periods.

All in all, it is evident that spatial interpolation with wrapped spatial Gaussian process models is dependent on several factors like hyperparameter and hyperprior selection, training sample sizes as well as the given characteristics of the observed directions in the training data like the level of dispersion.
At the same time, wrapped spatial Gaussian process models provide a powerful tool for directional spatial data analysis in a Bayesian framework as they explicitly allow to model prior assumptions over functions with spatial and temporal dependence structures on a circular domain and are able to interpolate directions better than assuming the mean direction when variability in the observed directions is high.
With the work from e.g. Jona Lasinio, Gelfand, and Jona Lasinio (2012), G. Jona Lasinio, Gelfand, and Jona Lasinio (2019) and Jona Lasinio, Santoro, and Mastrantonio (2020), extensive resources for wrapped spatial Gaussian process models are available and `R`-packages like `CircSpaceTime` (G. Jona Lasinio, Mastrantonio, and Santoro 2019) allow to implement and investigate these models straightforwardly.
Starting from the results presented here, it is the task of future work to drive forward the formulated ideas and hypotheses and to further investigate robustness and the influential factors on spatial interpolation accuracy of wrapped spatial Gaussian process models.

References

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2020. *Rmarkdown: Dynamic Documents for R*. https://github.com/rstudio/rmarkdown.

Breckling, Jens. 1989. *The Analysis of Directional Time Series: Application to Wind Speed and Direction*. Lecture Notes in Statistics, Vol 61. Springer-Verlag, Berlin.

Breitenberger, Ernst. 1963. "Analogues of the Normal Distribution on the Circle and the Sphere." *Biometrika* 50 (1-2): 81–88. https://doi.org/10.1093/biomet/50.1-2.81.

Clark, Michael. 2020. "Model Estimation by Example: Gaussian Processes." https://m-clark.github.io/models-by-example/gaussian-process.html.

Coles, Stuart. 1998. "Inference for Circular Distributions and Processes." *Statistics and Computing* 8 (2): 105–13. https://doi.org/10.1023/A:1008930032595.

Coles, Stuart, and Edward Casson. 1998. "Extreme Value Modeling for Hurricane Wind Speeds." *Structural Safety* 20 (3): 283–96. https://doi.org/10.1016/S0167-4730(98)00015-0.

Diggle, Peter J., and Paulo J. Ribeiro jr. 2007. *Model-Based Geostatistics*. Springer Series in Statistics. Springer Science+Business Media, LLC. https://doi.org/10.1007/978-0-387-48536-2.

Feng, Miao, Weimin Zhang, Xiangru Zhu, Boheng Duan, Mengbin Zhu, and De Xing. 2018. "Multivariate Interpolation of Wind Field Based on Gaussian Process Regression." *Atmosphere* 9 (5). MDPI AG: 194. https://doi.org/10.3390/atmos9050194.

Fernández i Marín, Xavier. 2020. *Ggmcmc: Tools for Analyzing Mcmc Simulations from Bayesian Inference*. https://CRAN.R-project.org/package=ggmcmc.

Fisher, Nicholas I. 1993. *Statistical Analysis of Circular Data*. Cambridge University Press. https://doi.org/10.1017/CBO9780511564345.

Fisher, Ronald Aylmer. 1953. "Dispersion on a Sphere." *Proceedings of the Royal Society of London, Series A* 217 (1130): 295–305. http://doi.org/10.1098/rspa.1953.0064.

Gebhardt, Hans, Rüdiger Glaser, Ulrich Radtke, Paul Reuber, and Andreas Vött, eds. 2020. *Geographie: Physische Geographie Und Humangeographie*. Springer Verlag GmbH.

Gelman, Andrew, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. 2014. *Bayesian Data Analysis*. 3rd ed. Texts in Statistical Science. Boca Raton: Chapman & Hall/CRC Press.

Gneiting, Tilmann. 2002. "Nonseparable, Stationary Covariance Functions for Space–Time Data." *Journal of the American Statistical Association* 97 (458). Taylor & Francis: 590–600. https://doi.org/10.1198/016214502760047113.

Gneiting, Tilmann, and Peter Guttorp. 2010. "Continuous Parameter Stochastic Process Theory." In *Handbook of Spatial Statistics*, edited by Alan E. Gelfand, Peter J. Diggle, Montserrat Fuentes, and Peter Guttorp (Eds.), 17–27. Handbooks of Modern Statistical Methods. Boca Raton: CRC Press. https://doi.org/10.1201/9781420072884.

Görtler, Jochen, Rebecca Kehlbeck, and Oliver Deussen. 2019. "A Visual Exploration of Gaussian Processes." *Distill Pub*. https://doi.org/10.23915/distill.00017.

Grimit, Eric P., Tilmann Gneiting, Veronica J. Berrocal, and N. A. Johnson. 2006. "The Continuous Ranked Probability Score for Circular Variables and Its Application to Mesoscale Forecast Ensemble Verification." *Quarterly Journal of the Royal Meteorological Society* 132 (621C). John Wiley & Sons, Ltd: 2925–42.

Haeseler, Susanne, Peter Bissolli, Christina Lefebvre, Jan Daßler, and Volker Zins. 2019. "Serie von Sturmtiefs Im März 2019 über Europa Mit Orkanböen in Deutschland." *Abteilung Klimaüberwachung*. Deutscher Wetterdienst.

Härdle, Wolfgang K., and Léopold Simar. 2019. *Applied Multivariate Statistical Analysis*. 5th ed. Springer International Publishing AG. https://doi.org/10.1007/978-3-662-45171-7.

Held, Leonhard, and Daniel Sabanés Bové. 2014. *Applied Statistical Inference: Likelihood and Bayes*. Springer Verlag, Berlin Heidelberg.

Henry, Lionel, and Hadley Wickham. 2020. *Purrr: Functional Programming Tools*. https://CRAN.R-project.org/package=purrr.

Jammalamadaka, S. Rao, and Ambar SenGupta. 2001. *Topics in Circular Statistics*. Series on Multivariate Analysis, Vol. 5. World Scientific Publishing Co. Pte. Ltd.

Jona Lasinio, Giovanna, Alan Gelfand, and Mattia Jona Lasinio. 2012. "Spatial Analysis of Wave Direction Data Using Wrapped Gaussian Processes." *The Annals of Applied Statistics* 6 (4). The Institute of Mathematical Statistics: 1478–98. https://doi.org/10.1214/12-AOAS576.

———. 2019. "Spatial and Spatio-Temporal Circular Processes with Application to Wave Directions." In *Applied Directional Statistics: Modern Methods and Case Studies*, edited by Christophe Ley and Thomas Verdebout (Eds.), 129–62. Interdisciplinary Statistics Series. Chapman & Hall/CRC Press, Taylor & Francis Group.

Jona Lasinio, Giovanna, Gianluca Mastrantonio, and Mario Santoro. 2019. *CircSpaceTime: Spatial and Spatio-Temporal Bayesian Model for Circular Data*. https://github.com/santoroma/CircSpaceTime.

Jona Lasinio, Giovanna, Mario Santoro, and Gianluca Mastrantonio. 2020. "CircSpaceTime: An R Package for Spatial and Spatio-Temporal Modelling of Circular Data." *Journal of Statistical Computation and Simulation* 90 (7). Taylor & Francis: 1315–45. https://doi.org/10.1080/00949655.2020.1725008.

Kent, John. 1978. "Limiting Behaviour of the von Mises-Fisher Distribution." *Mathematical Proceedings of the Cambridge Philosophical Society* 84 (3). Cambridge University Press: 531–36.

Klein, Nadja. 2019. "Lecture Slides in Introduction to Statistical and Machine Learning." Humboldt-Universität zu Berlin, Chair of Statistics.

Kruschke, John K. 2015. *Doing Bayesian Data Analysis*. 2nd ed. Boston: Academic Press. https://doi.org/10.1016/B978-0-12-405888-0.09993-1.

Kuhn, Max, Fanny Chow, and Hadley Wickham. 2020. *Rsample: General Resampling Infrastructure*. https://CRAN.R-project.org/package=rsample.

Lang, Moritz N., Georg J. Mayr, Reto Stauffer, and Achim Zeileis. 2019. "Bivariate Gaussian Models for Wind Vectors in a Distributional Regression Framework." *Advances in Statistical Climatology Meteorology and Oceanography* 5 (2): 115–32. https://doi.org/10.5194/ascmo-5-115-2019.

Latif, Mojib. 2009. *Klimawandel Und Klimadynamik*. Eugen Ulmer KG, Stuttgart.

Lee, Alan. 2010. "Circular Data." *WIREs Computational Statistics* 2 (4): 477–86. https:

//doi.org/10.1002/wics.98.

Lee, Sunbok, Suppanut Sriutaisuk, and Hanjoe Kim. 2020. "Using the Tidyverse Package in R for Simulation Studies in Sem." *Structural Equation Modeling: A Multidisciplinary Journal* 27 (3). Routledge: 468–82. https://doi.org/10.1080/10705511.2019.1644515.

Ley, Christophe, and Thomas Verdebout. 2017. *Modern Directional Statistics*. Interdisciplinary Statistics Series. Chapman & Hall/CRC Press, Taylor & Francis Group.

———, eds. 2019. *Applied Directional Statistics: Modern Methods and Case Studies*. Interdisciplinary Statistics Series. Chapman & Hall/CRC Press, Taylor & Francis Group.

Lund, Ulric, and Claudio Agostinelli. 2018. *CircStats: Circular Statistics, from "Topics in Circular Statistics" (2001)*. https://CRAN.R-project.org/package=CircStats.

Lund, Ulric, Claudio Agostinelli, Hiroyoshi Arai, Alessando Gagliardi, Eduardo Garcia Portugues, Dimitri Giunchi, Jean-Olivier Irisson, Matthew Pocernich, and Federico Rotolo. 2017. *Circular: Circular Statistics*. https://CRAN.R-project.org/package=circular.

Mardia, Kanti V., and Peter E. Jupp. 2000. *Directional Statistics*. Wiley Series in Probability and Statistics. John Wiley & Sons, Ltd.

Mastrantonio, Gianluca, Alan E. Gelfand, and Giovanna Jona Lasinio. 2016. "The Wrapped Skew Gaussian Process for Analyzing Spatio-Temporal Data." *Stochastic Environmental Research and Risk Assessment* 30 (8): 2231–42. https://doi.org/10.1007/s00477-015-1163-9.

Mastrantonio, Gianluca, Giovanna Jona Lasinio, and Alan E. Gelfand. 2016. "Spatio-Temporal Circular Models with Non-Separable Covariance Structure." *TEST* 25 (2): 331–50. https://doi.org/10.1007/s11749-015-0458-y.

Murphy, Kevin P. 2012. *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning. The MIT Press, Cambridge, Massachusetts.

Nguyen, Phi. 2020. *Huwiwidown: An Rmarkdown Thesis Template for Students at the Humboldt Universitaet Zu Berlin Wirtschaftswissenschaftlichen Fakultaet (School of Business and Economics) Using the 'Bookdown' Package.*

Peikert, Aaron. 2019. "Maketest: Testing a Flexible Parallelized Approach to Execute R Scripts via Make." *GitHub Repository*. https://github.com/aaronpeikert/maketest;

GitHub.

Pidwirny, M., and S. Jones. 2010. *Fundamentals of Physical Geography*. Ebooks on Physical Geography. University of British Columbia Okanagan. http://www.physicalgeography.net/fundamentals/chapter7.html.

Plummer, Martyn, Nicky Best, Kate Cowles, Karen Vines, Deepayan Sarkar, Douglas Bates, Russell Almond, and Arni Magnusson. 2020. *Coda: Output Analysis and Diagnostics for Mcmc*. https://CRAN.R-project.org/package=coda.

Rasmussen, Carl Edward, and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. The MIT Press, Cambridge, Massachusetts.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

RStudio Team. 2019. *RStudio: Integrated Development Environment for R*. Boston, MA: RStudio, Inc. http://www.rstudio.com/.

Shi, Yuge. 2019. "Gaussian Processes, Not Quite for Dummies." *The Gradient*. https://thegradient.pub/gaussian-process-not-quite-for-dummies/.

Stephens, M. A. 1963. "Random Walk on a Circle." *Biometrika* 50 (3/4): 385–90. http://www.jstor.org/stable/2333907.

Toulkeridis, Theofilos, and Imme Zach. 2017. "Wind Directions of Volcanic Ash-Charged Clouds in Ecuador - Implications for the Public and Flight Safety." *Geomatics, Natural Hazards and Risk* 8 (2). Taylor & Francis: 242–56. https://doi.org/10.1080/19475705.2016.1199445.

Turner, Richard E. 2016. "Lecture and Tutorial Slides on Gaussian Processes: From the Basics to the State-of-the-Art." http://cbl.eng.cam.ac.uk/pub/Public/Turner/News/imperial-gp-tutorial.pdf; Computational; Biological Learning Lab, Department of Engineering, University of Cambridge.

Wang, Fangpo, and Alan E. Gelfand. 2014. "Modeling Space and Space-Time Directional Data Using Projected Gaussian Processes." *Journal of the American Statistical Association* 109 (508): 1565–80. https://doi.org/10.1080/01621459.2014.934454.

Wang, Fangpo, Alan E. Gelfand, and Giovanna Jona Lasinio. 2015. "Joint Spatio-Temporal Analysis of a Linear and a Directional Variable: Space-Time Modeling of Wave Heights and Wave Directions in the Adriatic Sea." *Statistica Sinica* 25 (1). Institute of Statistical Science, Academia Sinica: 25–39. http://www.jstor.org/stable/24311002.

Watson, G. S. 1961. "Goodness-of-Fit Tests on a Circle." *Biometrika* 48 (1/2): 109–14. https://doi.org/10.1093/biomet/48.1-2.109.

Wickham, Hadley. 2019. *Tidyverse: Easily Install and Load the Tidyverse*. https://CRAN.R-project.org/package=tidyverse.

———. 2020. *Tidyr: Tidy Messy Data*. https://CRAN.R-project.org/package=tidyr.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2020. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Xie, Yihui. 2020a. *Bookdown: Authoring Books and Technical Documents with R Markdown*. https://github.com/rstudio/bookdown.

———. 2020b. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://yihui.org/knitr/.

A Appendix

A.1 Boxplots for mean CRPS in extended simulation setup

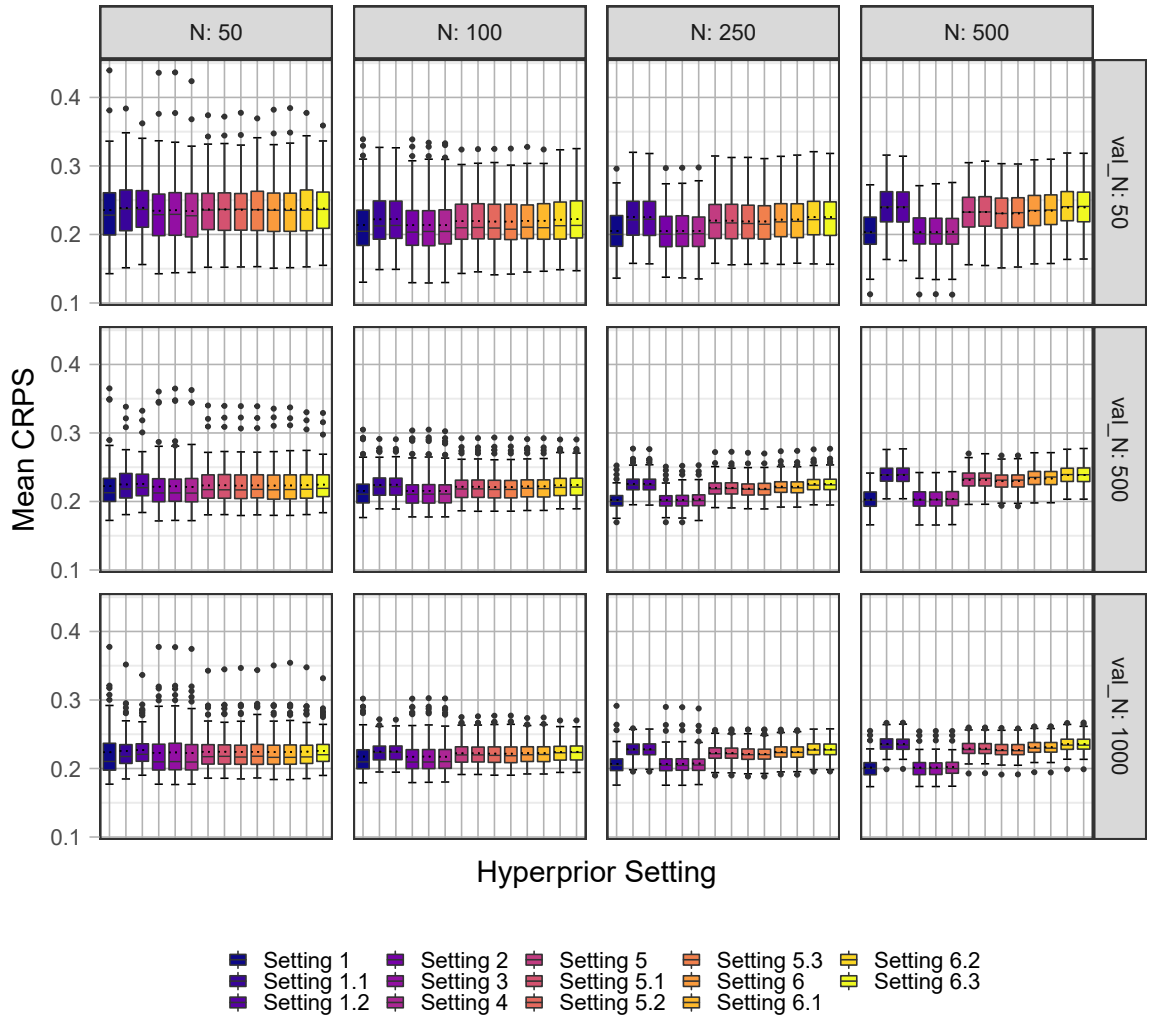


Figure 16: Boxplots of mean CRPS for the extended simulation setup. Each boxplot summarizes mean CRPS for the 100 replicates in each of the combinations of hyperprior settings and training and test sample sizes. Median values as solid lines, mean values as dashed lines.

A.2 Overall mean APE in extended simulation setup

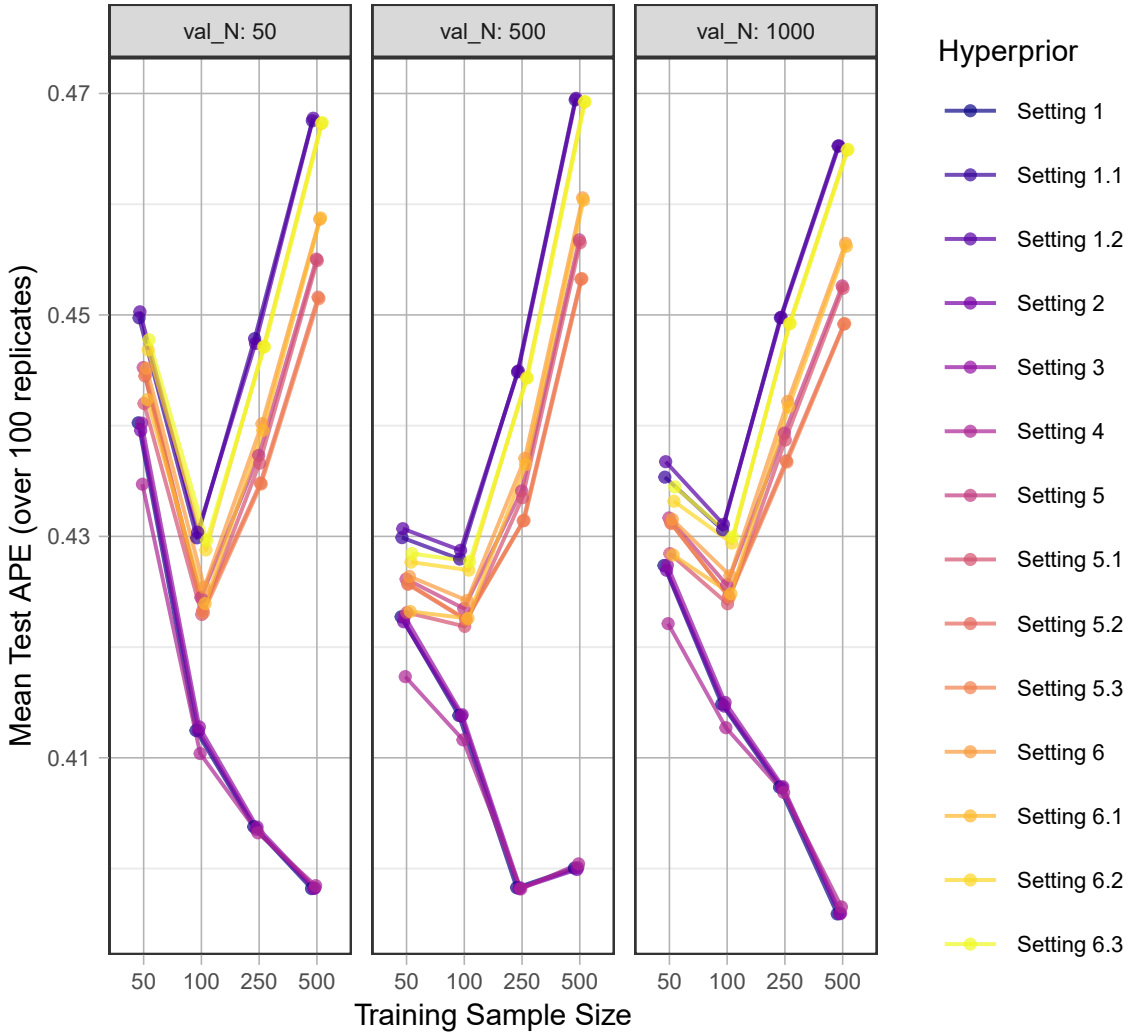


Figure 17: Mean APE over all 100 replicates in extended simulation setup.

A.3 Overall mean CRPS in extended simulation setup

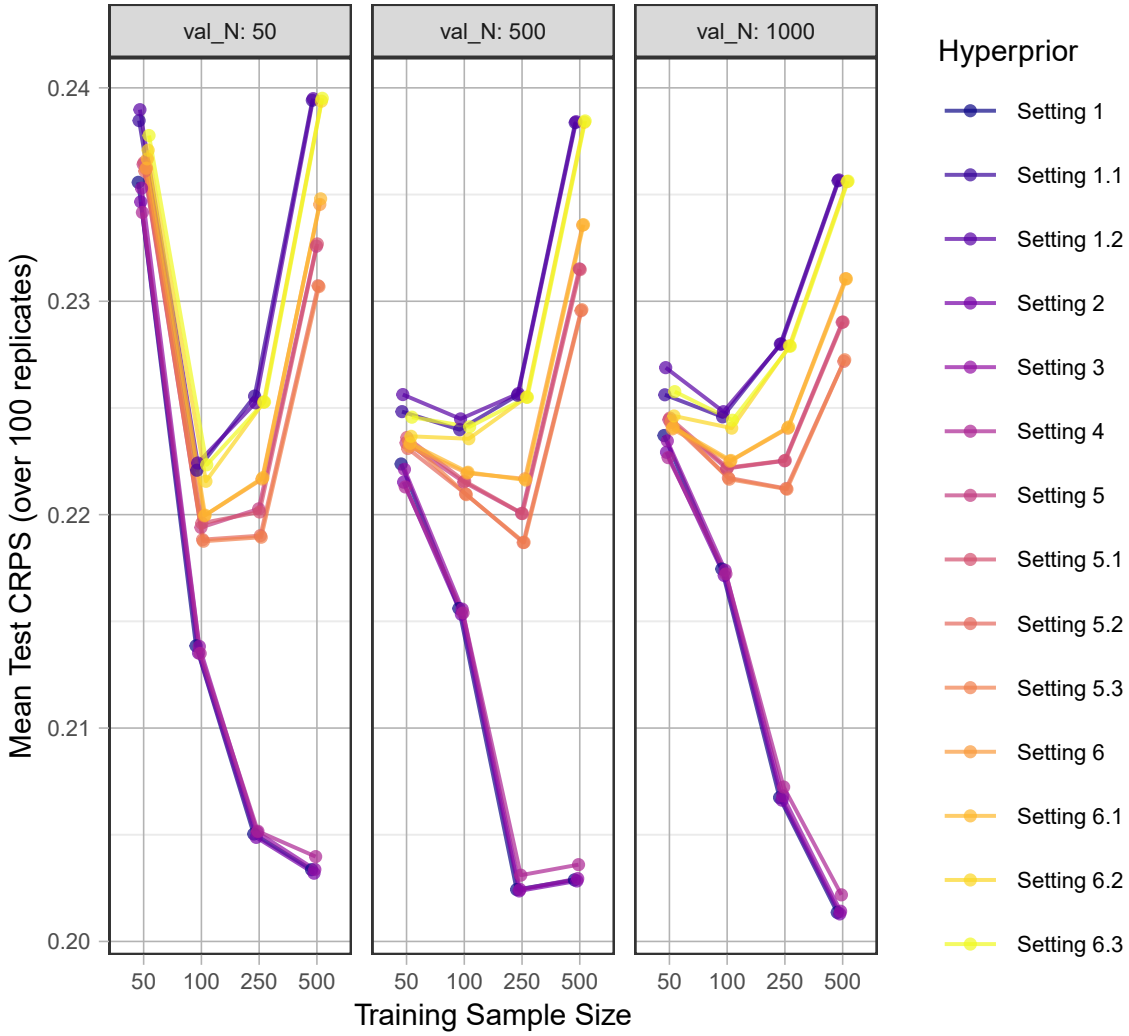


Figure 18: Mean CRPS over all 100 replicates in extended simulation setup.

A.4 Traceplots for two different test locations

Providing additional convergence checks for models in Chapter 4.

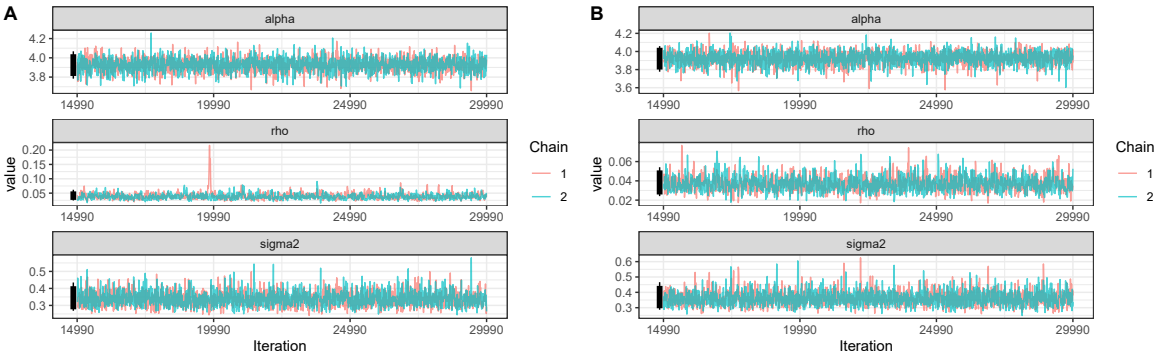


Figure 19: Traceplots of the model parameters of the wrapped Gaussian process models for wind directions at two different sets of training and test locations on 04.03.2019. Models are fitted with two MCMC chains with burn-in = 15000 and thinning = 10.

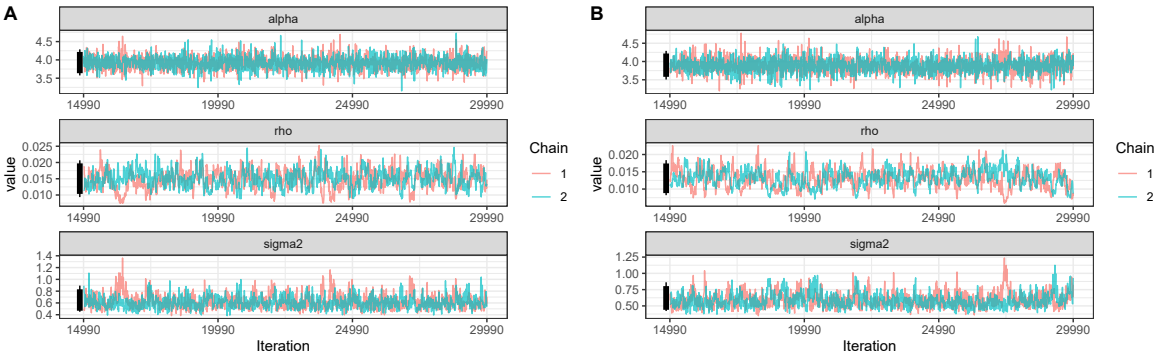


Figure 20: Traceplots of the model parameters of the wrapped Gaussian process models for wind directions at two different sets of training and test locations on 10.03.2019. Models are fitted with two MCMC chains with burn-in = 15000 and thinning = 10.

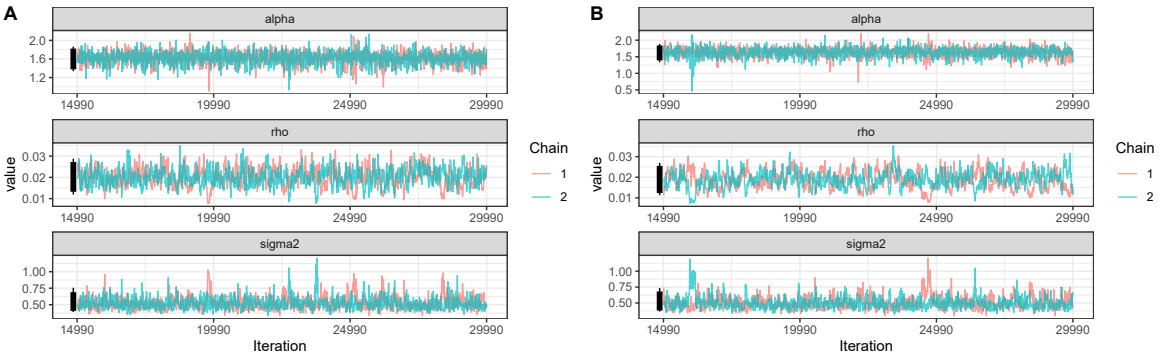


Figure 21: Traceplots of the model parameters of the wrapped Gaussian process models for wind directions at two different sets of training and test locations on 24.06.2019. Models are fitted with two MCMC chains with burn-in = 15000 and thinning = 10.

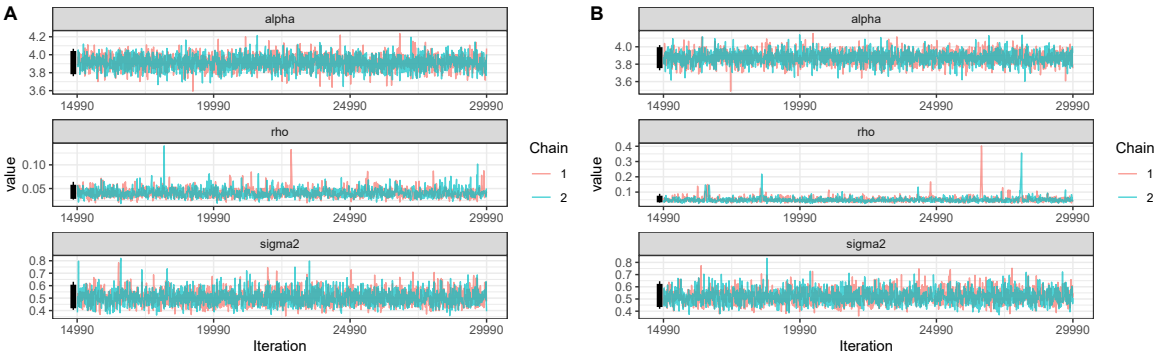


Figure 22: Traceplots of the model parameters of the wrapped Gaussian process models for wind directions at two different sets of training and test locations on 30.06.2019. Models are fitted with two MCMC chains with burn-in = 15000 and thinning = 10.

A.5 Plots of running means for two different test locations

Some text.

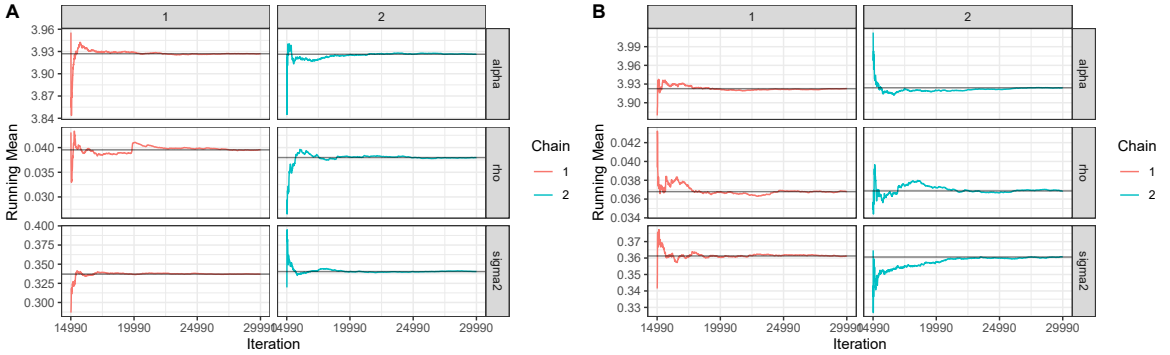


Figure 23: Plot of running means of the model parameters of the wrapped Gaussian process models for wind directions at two different sets of training and test locations on 04.03.2019. Results are obtained with two MCMC chains with burn-in = 15000 and thinning = 10.

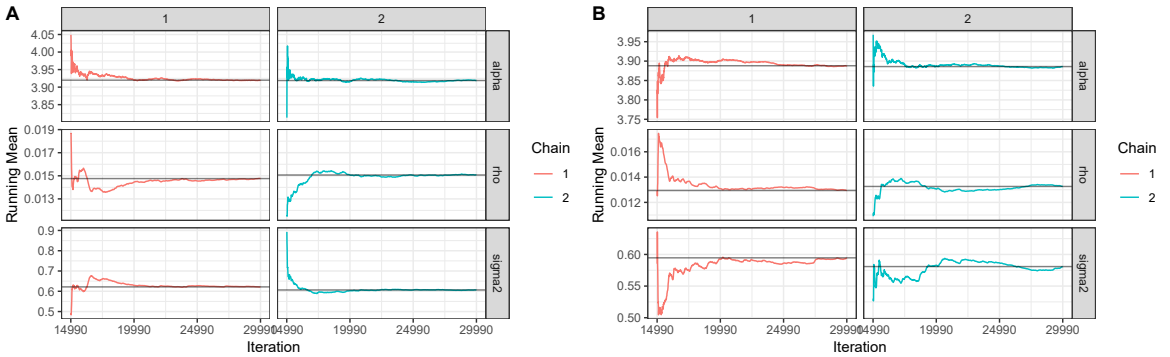


Figure 24: Plot of running means of the model parameters of the wrapped Gaussian process models for wind directions at two different sets of training and test locations on 10.03.2019. Results are obtained with two MCMC chains with burn-in = 15000 and thinning = 10.

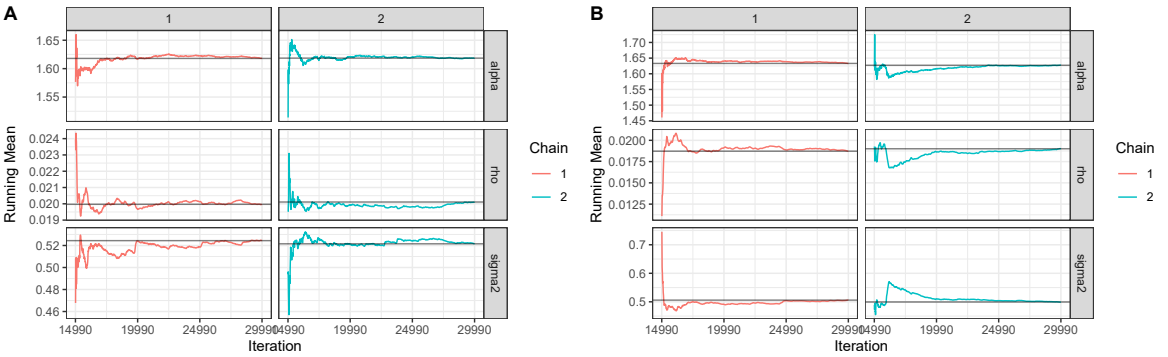


Figure 25: Plot of running means of the model parameters of the wrapped Gaussian process models for wind directions at two different sets of training and test locations on 24.06.2019. Results are obtained with two MCMC chains with burn-in = 15000 and thinning = 10.

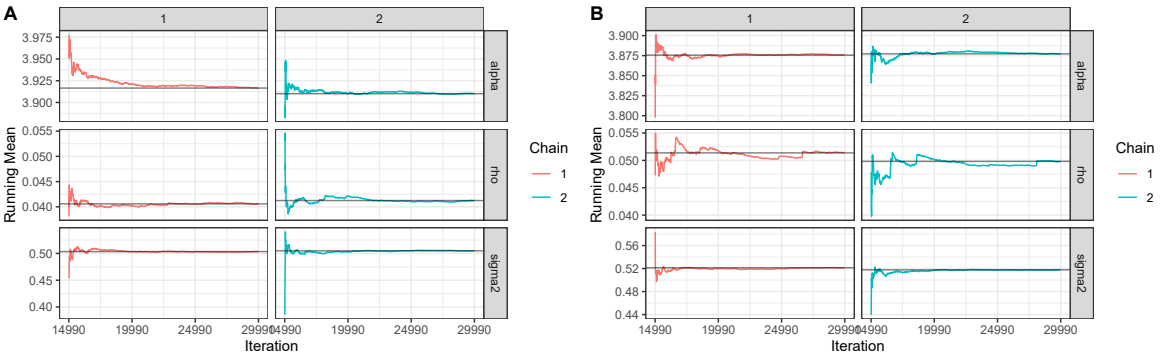


Figure 26: Plot of running means of the model parameters of the wrapped Gaussian process models for wind directions at two different sets of training and test locations on 30.06.2019. Results are obtained with two MCMC chains with burn-in = 15000 and thinning = 10.

A.6 Predicted wind directions on 04.03.2019 and 24.06.2019

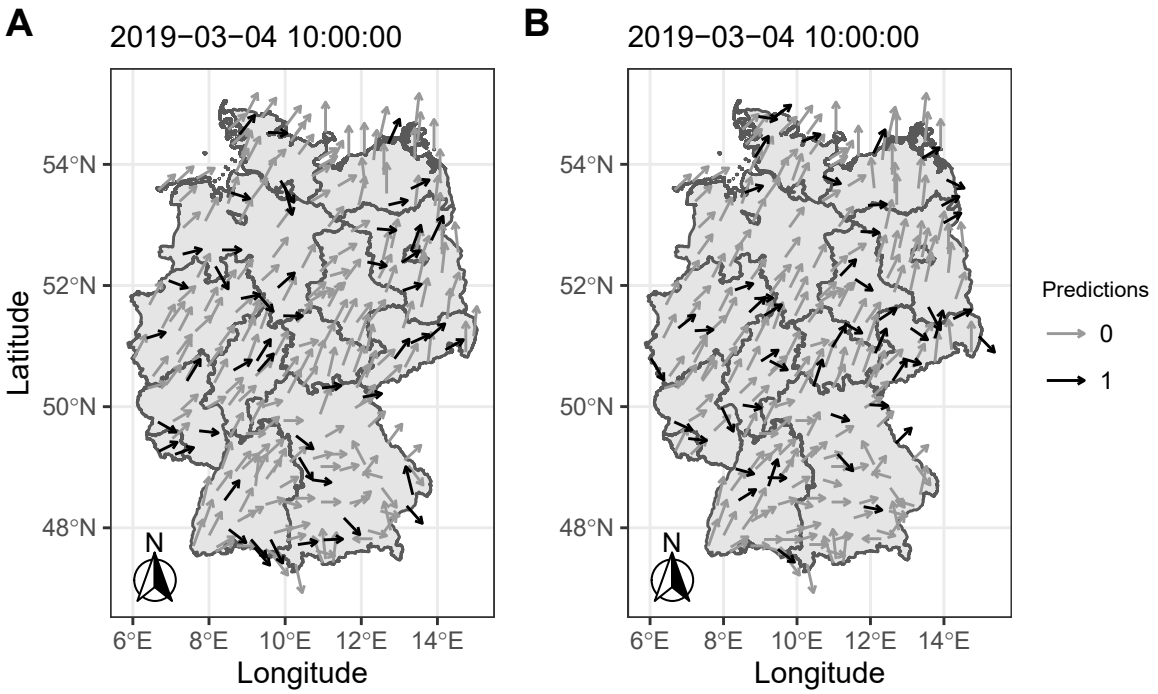


Figure 27: Predicted wind directions (black arrows) at first (A) and second (B) set of test locations for data observed at 10:00:00 AM on 04.03.2019. Training observations depicted as grey arrows.

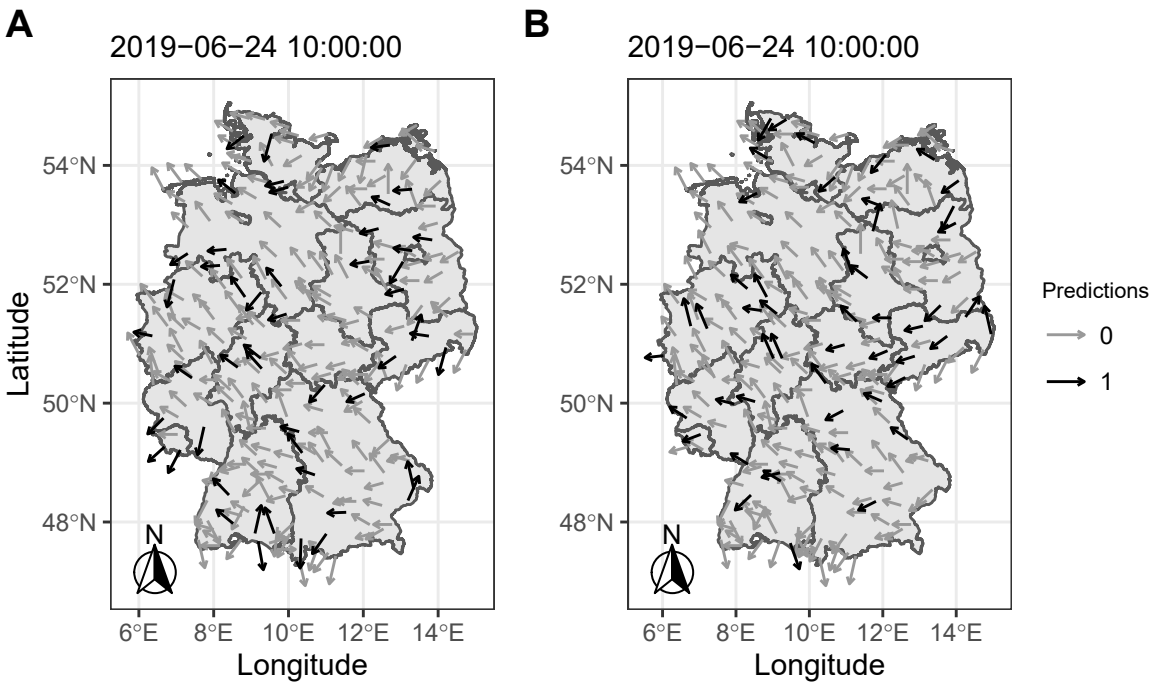


Figure 28: Predicted wind directions (black arrows) at first (A) and second (B) set of test locations for data observed at 10:00:00 AM on 24.06.2019. Training observations depicted as grey arrows.

A.7 Tables

This table presents averaged mean and modal point estimates, credible intervals and mean APE and CRPS on test data with sample size `val_N` = 500 for the exploratory setup in section 3.5.1.

	mean $\hat{\mu}$		concentration $\hat{c}$		spatial decay $\hat{rho}$		for test sample with `val_N` = 500	
N	m	covrate	m	covrate	m	covrate	APE	CRPS
Setting 1								
50	3.142960	0.95	0.7768969	0.99	4.9190778	0.03	0.4226852	0.2222398
100	3.138825	0.88	0.7781223	0.94	5.0259421	0.01	0.4139797	0.2156614
250	3.138486	0.87	0.7803772	0.88	4.8241528	0.00	0.3982052	0.2023962
500	3.142018	0.71	0.7784348	0.92	5.0164053	0.00	0.4000632	0.2029035
Setting 2								
50	3.140469	0.93	0.7767925	0.99	2.1375950	0.03	0.4220769	0.2215481
100	3.137927	0.88	0.7779857	0.96	2.4662074	0.01	0.4136273	0.2152218
250	3.138386	0.87	0.7803417	0.90	2.5663711	0.00	0.3981407	0.2023660
500	3.141739	0.71	0.7783640	0.91	1.8137820	0.00	0.3999449	0.2028029
Setting 3								
50	3.142969	0.93	0.7767274	0.99	0.9409965	0.08	0.4224274	0.2219652
100	3.139003	0.90	0.7779230	0.93	0.9825623	0.01	0.4138948	0.2155217
250	3.138513	0.88	0.7802164	0.87	1.2725552	0.00	0.3983507	0.2025265
500	3.141817	0.72	0.7782912	0.90	1.0438646	0.00	0.4000635	0.2029458
Setting 4								
50	3.143014	0.93	0.7818456	0.98	0.2685214	0.21	0.4171771	0.2212315
100	3.139689	0.89	0.7801764	0.93	0.2411867	0.06	0.4114464	0.2154181
250	3.138416	0.88	0.7802988	0.88	0.3845072	0.00	0.3981021	0.2031102
500	3.142221	0.73	0.7774789	0.91	0.3914450	0.00	0.4004117	0.2035548
Setting 5								
50	3.142191	0.97	0.7602507	0.93	0.0272674	1.00	0.4263381	0.2234102
100	3.148189	0.92	0.7544093	0.87	0.0268946	0.82	0.4235322	0.2215325
250	3.140754	0.96	0.7190456	0.31	0.0292736	0.00	0.4340459	0.2201188
500	3.152525	0.91	0.6705630	0.00	0.0296601	0.00	0.4568066	0.2315416
Setting 6								
50	3.143004	0.97	0.7614400	0.91	0.0248638	1.00	0.4263513	0.2231660
100	3.147765	0.90	0.7525073	0.83	0.0235316	0.88	0.4243347	0.2219458
250	3.140487	0.96	0.7124492	0.23	0.0268521	0.00	0.4370706	0.2216681
500	3.152304	0.97	0.6596772	0.00	0.0276428	0.00	0.4606477	0.2335959

Table 11: Results from simulated data in exploratory setup. Posterior mean estimates (m) obtained by averaging over MCMC samples and 100 replicates for each setting, decay parameter as modal value over MCMC samples and replicates. Coverage rates (covrate) give percentage of true values contained in 95%-credible intervals of the given posterior. True values: alpha = π, concentration = 0.7788 and spatial decay = 0.021.

Declaration of Authorship

I hereby confirm that I have authored this Master Thesis independently and without use of other than the indicated sources. All passages which are literally or in general matter taken out of publications or other sources are marked as such.

Berlin, 23 November 2020

. . . . . .

Anna Elisabeth Riha